



PROYECTO FIN DE CARRERA PLAN 2000

E.U.I.T. TELECOMUNICACIÓN

TEMA: Internet Móvil

TÍTULO: Control de módulo 3G mediante aplicación Android

AUTOR: David Okumura Bran

TUTOR: Antonio da Silva Fariña

DEPARTAMENTO: DIATEL

Miembros del Tribunal Calificador:

PRESIDENTE: Pedro García Pino

VOCAL: Antonio da Silva Fariña

Vº Bº.

VOCAL SECRETARIO: Carlos Ramos Nespereira

DIRECTOR:

Fecha de lectura:

Calificación:

El Secretario,

RESUMEN DEL PROYECTO:

Este proyecto describe la metodología a seguir para conectar la plataforma Arduino a dispositivos Android y establecer una conexión que permita controlar dicha plataforma. Sobre Arduino se acoplará un módulo 3G que permitirá hacer uso de funcionalidades propias de los teléfonos móviles.

El objetivo final del proyecto era el control del módulo 3G mediante comandos AT enviados desde un dispositivo Android (tableta) conectado a través de USB. Para ello, se ha desarrollado una aplicación de demostración que permite el uso de algunas de las funcionalidades de comunicación del módulo 3G.

El proyecto proporciona una guía de iniciación con explicaciones de los diferentes dispositivos, tecnologías y pasos a seguir para la integración de las diferentes plataformas que se han usado en el proyecto: Arduino, Módulo de comunicaciones 3G, y Android.





Escuela Universitaria de
Ingeniería Técnica de
Telecomunicación



Proyecto Fin de Carrera

Control de módulo 3G mediante aplicación Android

Autor: David Okumura Bran

Tutor: Antonio da Silva Fariña







Resumen

Este proyecto describe la metodología a seguir para conectar la plataforma Arduino a dispositivos Android y establecer una conexión que permita controlar dicha plataforma. Sobre Arduino se acoplará un módulo 3G que permitirá hacer uso de funcionalidades propias de los teléfonos móviles.

El objetivo final del proyecto era el control del módulo 3G mediante comandos AT enviados desde un dispositivo Android (tableta) conectado a través de USB. Para ello, se ha desarrollado una aplicación de demostración que permite el uso de algunas de las funcionalidades de comunicación del módulo 3G.

Para alcanzar el objetivo propuesto se ha investigado sobre temas tales como: internet de las cosas, las tecnologías de comunicaciones móviles, el sistema operativo Android y el desarrollo de aplicaciones móviles, la plataforma Arduino, el funcionamiento del módulo 3G y sobre la comunicación serie que permitirá comunicarse entre Android y módulo 3G.

El proyecto proporciona una guía de iniciación con explicaciones de los diferentes dispositivos, tecnologías y pasos a seguir para la integración de las diferentes plataformas que se han usado en el proyecto: Arduino, Módulo de comunicaciones 3G, y Android.







Abstract

This project describes the methodology to connect the Arduino platform to Android devices and establish a connection to allow the platform control. A 3G module will be engaged on Arduino allowing the usage of mobile phones functionalities.

The main objective of the project was the control of 3G module through AT commands sent from an Android device (tablet) connected via USB. For that, a demonstration application was developed to permit the use of some communication features of 3G module.

To achieve the target, an investigation has been carried out about issues such as: internet of things, mobile communications technologies, the Android operating system and mobile applications development, the Arduino platform, the 3G module operation and serial communication that allows the communication between Android and the 3G module.

The project provides a starter guide with explanations of the different devices, technologies and steps for the integration of the different platforms that have been used in the project: Arduino, 3G communications module and Android.







Agradecimientos

A mi Madre.

Y a mis abuelos.

Porque se lo debo todo.

GRACIAS.







Índice de contenidos

Resumen	5
Abstract.....	7
Agradecimientos	9
Índice de contenidos	11
Índice de Figuras.....	13
1. Introducción.....	17
1.1. Motivación del proyecto	17
1.2. Planteamiento y objetivo	17
1.3. Estructura de la memoria	18
2. Estado del arte	19
2.1. Internet de las cosas	19
2.2. Tecnologías de comunicaciones móviles.....	20
2.2.1. Organismos	20
2.2.2. Evolución de los sistemas móviles.....	21
2.3. Android	25
2.3.1. Descripción	25
2.3.2. Entorno.....	26
2.3.3. Lenguaje.....	28
2.3.4. Código y librerías.....	31
2.4. Arduino	36
2.4.1. Descripción	36
2.4.2. Entorno de desarrollo para Arduino.....	40
2.4.3. Lenguaje.....	41
2.4.4. Código.....	42
2.5. Módulo 3G.....	43
2.5.1. Descripción	43
2.5.2. Comandos AT (ATtention)	47
2.6. Comunicación serie.....	47
3. Fases de desarrollo	49
3.1. Primera fase. Control de Arduino.	49





3.1.1.	Requisitos.....	49
3.1.2.	Instrucciones	49
3.1.3.	Resultado final	50
3.2.	Segunda fase. Control de Módulo 3G.....	50
3.2.1.	Requisitos.....	50
3.2.2.	Instrucciones	51
3.2.3.	Resultado final	54
3.2.4.	Inconvenientes encontrados	60
4.	Conclusión.....	63
5.	Líneas futuras de desarrollo.....	63
6.	Apéndice I (Relativo a Arduino)	65
A.	Como saber si el dispositivo Android tiene soporte USB Host Mode.....	65
B.	Cable USB	66
C.	Modos USB Host y Accesory de Android.....	68
D.	Guía de instalación de ADB drivers en PC.....	69
E.	Guía de instalación de Arduino en Windows	70
F.	Guía de instalación de Android en Windows	74
7.	Apéndice II (Relativo al Módulo 3G).....	75
G.	Instalación del Módulo 3G para Windows 8	75
H.	Introducción a HyperTerminal.....	80
I.	Configuraciones.....	85
J.	Guía de comandos AT	93
	Referencias	99





Índice de Figuras

Figura 1: Objetivo del proyecto	18
Figura 2: Internet of Things	19
Figura 3: Evolución IoT. Fuente: Cisco IBSG	19
Figura 4: Evolución de las tecnologías móviles	21
Figura 5: Despliegue de LTE. (5)	23
Figura 6: Gráfica comparativa de velocidades	24
Figura 7: Tabla de estándares de comunicaciones móviles (6)	25
Figura 8: Logo de Android	25
Figura 9: Logo Eclipse.....	26
Figura 10: Logo ADT	26
Figura 11: Interfaz Eclipse.....	27
Figura 12: LogCat de Eclipse	27
Figura 13: Imagen propia del proyecto en cuestión.....	28
Figura 14: Estructura de carpetas Android	29
Figura 15: Logo Physicaloid.....	31
Figura 16: Conexión USB serie	32
Figura 17: Logo Arduino	36
Figura 18: Placa Arduino UNO	37
Figura 19: Modelos de Arduino.....	39
Figura 20: Shields apilados.....	40
Figura 21: Entorno Arduino.....	41
Figura 22: Módulo3G	43
Figura 23: Módulo SIM5218E.....	43
Figura 24: Módulo 3G con sus accesorios.....	44
Figura 25: Diagrama de conexiones del Módulo 3G (Por delante)	45
Figura 26: Diagrama de conexiones del Módulo 3G (Por detrás)	45
Figura 27: Tableta con AppArduino conectada a Arduino.....	50
Figura 28: Tableta Android - Módulo 3G montado en Arduino	51
Figura 29: Icono USB Serial Monitor Lite	52
Figura 30: Captura configuración de USB Serial Monitor Lite	52
Figura 31: Captura funcionamiento de USB Serial Monitor Lite.....	53
Figura 32: Icono UsbTerminal.....	53
Figura 33: Resultado final del proyecto.....	54
Figura 34: AppMódulo3G (Menú principal)	54
Figura 35: AppMódulo3G (Menú principal conectado)	55
Figura 36: AppMódulo3G (Llamadas)	55
Figura 37: AppMódulo3G (Recibiendo llamada).....	56
Figura 38: AppMódulo3G (SMS).....	56
Figura 39: AppMódulo3G (Envío de SMS)	57





Figura 40: AppMódulo3G (SMS enviado)	57
Figura 41: AppMódulo3G (SMS recibidos)	58
Figura 42: AppMódulo3G (SMS recibidos remarcados).....	58
Figura 43: AppMódulo3G (Monitorización)	59
Figura 44: AppMódulo3G (Información de celdas)	59
Figura 45: AppMódulo3G (Consola).....	60
Figura 46: Logo USB OTG	66
Figura 47: Adaptador OTG (micro USB-USB tipo A).....	66
Figura 48: Tipos de USB	66
Figura 49: Android conectado con OTG a Arduino (25)	67
Figura 50: Imagen de la conexión Arduino – tableta vía USB	67
Figura 51: Modos USB Host y Accesory	68
Figura 52: Arduino UNO	70
Figura 53: Cable estándar USB	70
Figura 54: Ejemplo de Sketch Blink.....	72
Figura 55: seleccionar placa Arduino	72
Figura 56: Opción cargar	73
Figura 57: Ubicación del SDK Manager en Eclipse.....	74
Figura 58: Módulo 3G enchufado directamente al PC	75
Figura 59: Dispositivos sin controlador en panel de control	76
Figura 60: Propiedades de Windows	76
Figura 61: Actualizar controlador.....	77
Figura 62: Buscar controlador	77
Figura 63: Ventanas de drivers instalados del Módulo 3G.....	78
Figura 64: Dispositivos del Módulo 3G instalados	79
Figura 65: Hyper Terminal	80
Figura 66: Nueva conexión (HyperTerminal)	81
Figura 67: Configuración (HyperTerminal)	81
Figura 68: Detalle del botón de encendido/apagado del Módulo	82
Figura 69: Error por ausencia de SIM (HyperTerminal)	82
Figura 70: Conexión correcta (HyperTerminal)	83
Figura 71: Información del Módulo 3G (HyperTerminal)	83
Figura 72: Envío de SMS (Hyper Terminal)	84
Figura 73: Conexión directa Módulo 3G-Tableta	85
Figura 74: Conexión del Módulo 3G insertado en Arduino a la tableta.....	85
Figura 75: Detalle de los jumpers del Módulo 3G	86
Figura 76: Diagrama del Módulo 3G.....	86
Figura 77: Arduino UNO sin el microcontrolador	87
Figura 78: Detalle del puente en Arduino.....	87
Figura 79: Sketch vacío	88



Figura 80: Módulo 3G sobre Arduino	88
Figura 81: Micrófono y auricular	89
Figura 82: Antena 2G/3G	89
Figura 83: Tarjeta SIM	89
Figura 84: Diagrama Módulo 3G (Audio).....	90
Figura 85: Módulo 3G con elementos de audio.....	90
Figura 86: Conectores SIM3218 para antenas	91
Figura 87: Antena enchufada al Módulo 3G	91
Figura 88: Teléfono de cartón.....	92
Figura 89: Secuencia de colocación del Módulo3G en teléfono de cartón	92







1. Introducción

1.1. Motivación del proyecto

La realización de Proyecto Fin de Carrera brinda una oportunidad perfecta para aplicar los conocimientos adquiridos durante el transcurso de la carrera y también para desarrollar y aprender otros nuevos teniendo siempre la libertad de poder hacerlo creativamente antes de introducirse en el mundo laboral.

Al comienzo del proyecto, la intención era realizar un trabajo centrado en el tema de las comunicaciones móviles, posteriormente tras madurar más el asunto de interés se determinó que sería un buen punto de partida investigar sobre dos materias. Una es el desarrollo de aplicaciones móviles, más en concreto las destinadas a Android, ya que es el sistema operativo cada vez más popular en dispositivos móviles y tabletas a nivel mundial y sobre todo por ser libre; y la otra es el ámbito relacionado con la plataforma Arduino, otra tecnología que recientemente poco a poco se está abriendo camino como fruto de su bajo coste, sencillez y libertad hardware.

Los conceptos propuestos, independientemente son muy interesantes por su temática actual, pero el atractivo principal del proyecto lo aporta la unión de estas dos tecnologías para proporcionar un nuevo punto de vista más global, esto supone un concepto innovador con abundantes futuros usos prácticos.

1.2. Planteamiento y objetivo

Con la ejecución de este proyecto se pretende profundizar en la unión de las tecnologías anteriormente citadas, el sistema operativo de telefonía móvil Android y la plataforma de electrónica Arduino.

Para llevar a cabo la unión existen varios medios disponibles, inalámbricamente (WiFi, Blue Tooth, radio...) o por cable con USB, éste último es el método que se va a emplear por ser más sencillo y económico.

Para profundizar más en el tema a las comunicaciones móviles se utilizará un módulo (shield) compatible con Arduino que dispone de tecnología 2G y 3G, de ahora en adelante Módulo 3G, que es capaz de prestar servicios característicos de los teléfonos móviles.

El objetivo principal es controlar el Módulo 3G desde un dispositivo Android vía cable USB, y para ello se utilizarán comandos AT a modo de instrucciones para manejar dicho módulo.

Se realizará una aplicación en Android de demostración muy rudimentaria que muestre las facilidades que puede proporcionar el Módulo 3G, en dicha aplicación se podrán realizar y recibir llamadas, enviar y recibir SMS y se podrá obtener información de la celda que actualmente brinda servicio y las vecinas.



El proyecto servirá como guía de iniciación con explicaciones de los diferentes dispositivos, tecnologías y pasos a seguir para todo aquel que quiera profundizar con la plataforma Arduino, el Módulo 3G, Android y su comunicación mediante cable USB.

Para que quede más claro, en la siguiente imagen se representa el sistema que se busca conseguir.

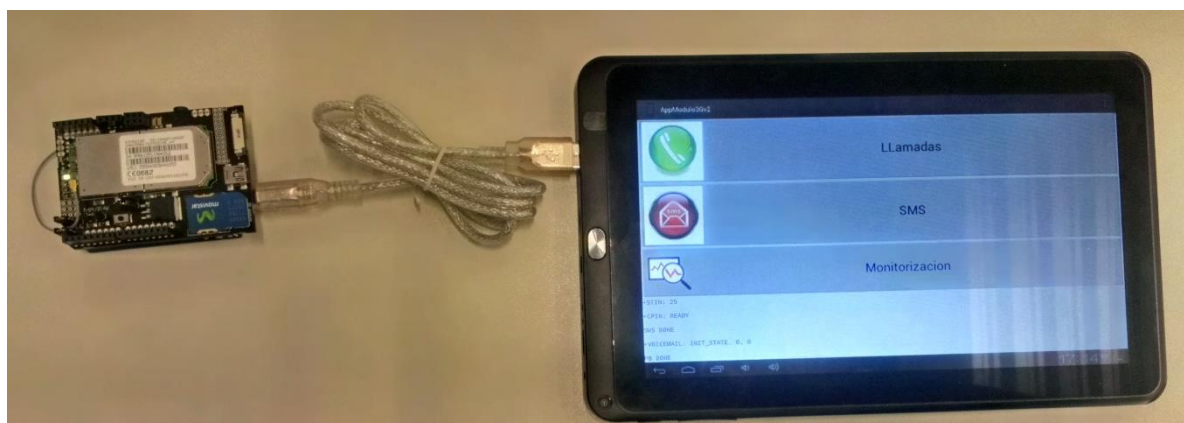


Figura 1: Objetivo del proyecto

1.3. Estructura de la memoria

La memoria se divide en tres apartados.

En primer lugar se presenta una base teórica donde se explican los conocimientos previos necesarios para realizar el proyecto tales como tecnologías y herramientas que se van a utilizar.

A continuación se expone el desarrollo donde se detallan los pasos a seguir, éste a su vez se divide en dos partes, primero se realiza una toma de contacto con Arduino para empezar con conceptos más sencillos y una vez controlada la plataforma Arduino se avanza a la siguiente fase con el Módulo 3G, donde se lleva a cabo la creación de una aplicación que controle mediante comandos AT las facilidades que otorga el módulo. Todo este desarrollo viene acompañado de una amplia documentación anexada a modo de guías de instalación y configuraciones.

Y por último se termina con las conclusiones obtenidas y los posibles usos futuros.



2. Estado del arte

2.1. Internet de las cosas



Figura 2: Internet of Things

Internet de las Cosas o IoT (Internet of Things) se basa en la conexión de objetos cotidianos a la Red con capacidad para percibir, procesar y transmitir datos entre ellos, creando una red de objetos interconectados entre sí con el objetivo de mejorar la calidad de vida (1).

Según datos de Cisco en 2008 menos del 1% de los objetos estaban conectados, pero considera que en 2015 habrá aproximadamente 25 mil millones de dispositivos conectados a internet y que en 2020 se duplicarán. (2)

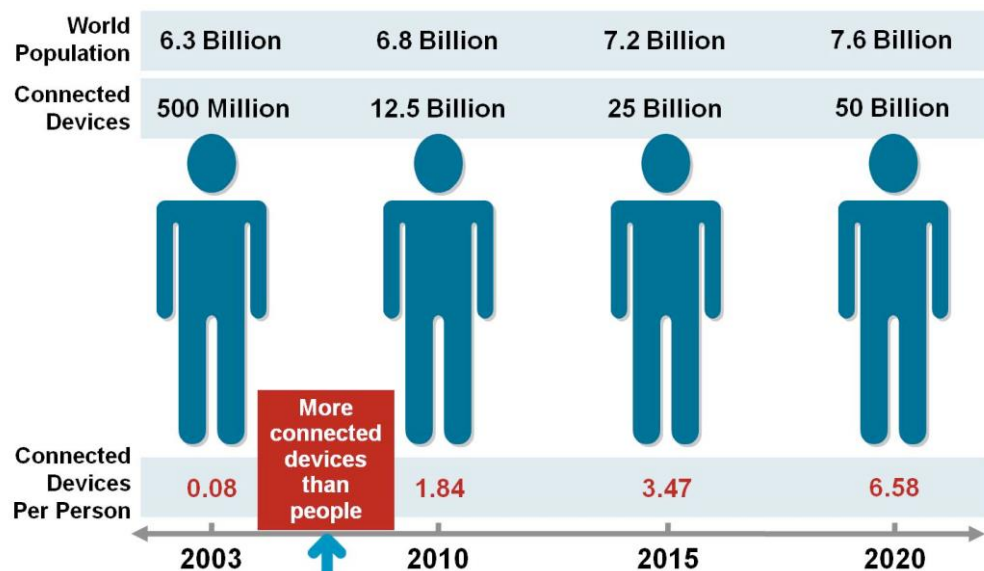


Figura 3: Evolución IoT. Fuente: Cisco IBSG

A día de hoy estos grandes avances en la conexión de los objetos a internet se deben a varios motivos:



- Cada dispositivo necesita una dirección IP y con la migración al protocolo IPv6, la cantidad de direcciones disponibles aumentó de sobra para conectar cada uno de los objetos que nos rodean.
- El coste de almacenamiento de todos los datos que estos objetos crearán se ha visto reducido por los numerosos servicios de almacenamiento de información en la nube que han surgido.

Pero para que todo esto sea posible son necesarias redes de comunicación inalámbrica M2M (Machine to Machine) seguras y potentes, también es necesario aplicar a los sistemas y a los objetos una inteligencia (“Smart”) para procesar los datos recogidos por los sensores y convertirlos en información útil, y por último se necesitan unos potentes sistemas de análisis que manipulen esos grandes conjuntos de datos (“Big data”).

Arduino es una plataforma de hardware y software libre ideal para experimentar con el mundo del Internet de las cosas, permite además de controlar dispositivos, la comunicación e interacción entre objetos, es sencillo, barato y ampliable ya que se dispone de una gran variedad de modelos, sensores y actuadores compatibles, debido a su gran éxito tiene una importante comunidad de usuarios en la que se comparten proyectos y librerías software que permiten la comunicación por cable e inalámbrica a Internet.

2.2. Tecnologías de comunicaciones móviles

2.2.1. Organismos

La Unión Internacional de Telecomunicaciones (UIT) (3), con siglas en inglés ITU de ahora en adelante, es el organismo especializado de Telecomunicaciones de la Organización de las Naciones Unidas encargado de regular las telecomunicaciones a nivel internacional entre las distintas administraciones y empresas operadoras. Se ha encargado de crear la clasificación por Generaciones de los sistemas móviles, los números decimales son una clasificación comercial, por ejemplo: EDGE es del tipo 2G según la ITU, pero comercialmente es la 2’75G.

El Proyecto Asociación de Tercera Generación o más conocido por el acrónimo inglés 3GPP (3rd Generation Partnership Project) es una colaboración de grupos de asociaciones de telecomunicaciones, conocidos como Miembros Organizativos. (4)

El objetivo inicial del 3GPP era asentar las especificaciones de un sistema global de comunicaciones de tercera generación 3G para móviles basándose en las especificaciones del sistema evolucionado "Global System for Mobile Communications" GSM dentro del marco del proyecto internacional de telecomunicaciones móviles 2000 de la Unión Internacional de Telecomunicaciones ITU. Más tarde el objetivo se amplió incluyendo el desarrollo y mantenimiento de:

- El Sistema Global de telecomunicaciones móviles GSM incluyendo las tecnologías de radio-acceso evolucionadas del GSM (cómo por ejemplo GPRS o el EDGE).





- Un sistema de tercera generación evolucionado y más allá del sistema móvil basado en las redes de núcleo evolucionadas del 3GPP y las tecnologías de radio-acceso apoyadas por los miembros del proyecto (cómo por ejemplo la tecnología UTRAN y sus modos FDD y TDD).
- Un Subsistema Multimedia IP (IMS) desarrollado en un acceso de forma independiente.

Los estándares europeos de telefonía móvil celular GSM, UMTS y LTE fueron desarrollados y son mantenidos por el 3GPP.

2.2.2. Evolución de los sistemas móviles

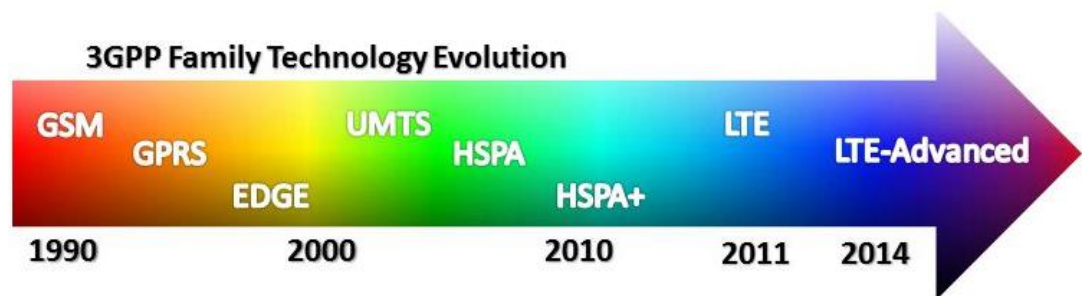


Figura 4: Evolución de las tecnologías móviles

0G: "Generación 0"

Los teléfonos móviles de esta generación eran unos radioteléfonos disponibles como un servicio comercial conectado a la red de telefonía fija, con sus propios números, normalmente ubicados en vehículos por su gran tamaño. La primera llamada fue en 1946.

1G: Tecnología analógica

Son sistemas celulares que solo soportan únicamente servicios de voz, su interfaz radio es analógica y su señalización digital. No existe un estándar. La primera llamada fue en 1973.

2G: GSM. Global System for Mobile communications

El Sistema Global para Comunicaciones Móviles es el primer estándar de telefonía móvil totalmente digital, impulsado por Europa. Ofrecen tanto servicios de voz como de datos.

Pero no es la única, ya que existen también tecnologías 2G distintas como IS-95 o D-AMPS. Aun así GSM ha conseguido imponerse y la mayoría de las redes mundiales 2G usan esta tecnología.

2.5G: GPRS. General Packet Radio Service

El Servicio General de Paquetes vía Radio es una evolución de GSM para la transmisión de datos mediante conmutación de paquetes, permite a las redes celulares una mayor velocidad y ancho de banda sobre el GSM.





Este sistema permite navegar por páginas WAP, Wireless Application Protocol (protocolo de aplicaciones inalámbricas). El pago en los servicios que ofrece este sistema corresponde con la cantidad de datos que son descargados.

Permite velocidades de transferencia de 56 a 114 kbps.

2'75G: EDGE. Enhanced Data Rates for GSM Evolution

Tasas de Datos Mejoradas para la evolución de GSM es una evolución de GPRS, utiliza un esquema de modulación y codificación alternativo.

Puede alcanzar velocidades desde 115 kbps hasta 384 kbps.

2'9G: EGPRS-2. EDGE Evolved

Es la extensión de EDGE que permite velocidades de hasta 1,2 Mbps para enlace descendiente y de hasta 474 kbps en enlace ascendiente por usuario.

3G: UMTS. Universal Mobile Telecommunications System

El Sistema universal de telecomunicaciones móviles define una nueva red de acceso radio con mayor ancho de banda.

Posee capacidades multimedia, una velocidad de acceso a Internet elevada, la cual también le permite transmitir audio y video en tiempo real; y una transmisión de voz con calidad equiparable a la de las redes fijas.

La tecnología de acceso en la interfaz radio utilizada en UMTS es WCDMA, o lo que es lo mismo CDMA en banda ancha (Wideband Code Division Multiple Access, en español Acceso Múltiple por División de Código de Banda Ancha) permite la descarga de datos a velocidades de hasta 2Mbps.

A menudo se utilizan los acrónimos WCDMA indistintamente con UMTS, HSDPA o HSUPA, pues todos son WCDMA.

3'5G: HSDPA. High Speed Downlink Packet Access

La tecnología HSDPA es la optimización de la tecnología espectral UMTS/WCDMA, mejorando la capacidad de transferencia de bajada hasta alcanzar tasas de 14,4 Mbps.

3'75G: HSUPA. High Speed Uplink Packet Access

La tecnología HSUPA es una evolución de HSDPA para optimizar la tasa de transferencia de subida hasta 5,76 Mbps.

3'9G: HSPA+. Evolved HSPA

Evolución de HSPA provee velocidades de hasta 168 Mbps de bajada y 22 Mbps de subida, a través de una técnica multi-antena conocida como MIMO (Multiple-Input Multiple-Output) y modulación 64-QAM o combinando múltiples celdas en una con la técnica Dual-Cell HSDPA.





3'99G: LTE. Long Term Evolution¹

Este estándar de comunicaciones móviles tiene una interfaz radio completamente nueva y supone una nueva generación respecto a UMTS.

Permite altas tasas de bits con baja latencia, velocidades de hasta 300 Mbps en DL (Enlace Descendiente) y hasta 75Mbps en UL (Enlace Ascendiente).

Los servicios LTE están basados solo en conmutación de paquetes (no puede gestionar SMS o llamadas con conmutación de circuitos, de eso se encargarán las redes GSM y demás) por lo tanto es fácil y barato desplegarlo por los operadores.

Evita la fragmentación de los terminales a nivel mundial por el tipo de duplexación ya que es compatible con FDD (Frequency Division Duplex) que utiliza varias zonas del espectro y TDD (Time Division Duplex) que ocupa una sola zona.

La red LTE se ha desplegado sólo en ciudades con más de 100.000 habitantes, en Madrid provincia únicamente se puede disfrutar del servicio en Madrid ciudad. A continuación se muestra el mapa del despliegue de cada operador.



Figura 5: Despliegue de LTE. (5)

¹ La clasificación de LTE como de 3ª Generación en vez de 4ª como a veces se menciona se justifica en el siguiente párrafo. (32)

En 2008 la ITU creó el comité IMT – Advanced para determinar lo que debería ser una tecnología móvil de cuarta generación, el principal requisito era: “Velocidades máximas de transmisión de datos mejoradas para admitir aplicaciones y servicios avanzados (como objetivo a los efectos de la investigación, se han establecido velocidades de 100 Mbit/s para una movilidad alta y de 1 Gbit/s para una movilidad baja)”, requisito que LTE no cumple.

Pero incluso así, el IMT – Advanced decidió permitir que los candidatos a ser evolucionados a 4G como LTE puedan publicitarse con 4G.

4G: LTE Advanced

Es una mejora del estándar LTE para adaptarse a las exigencias del comité IMT – Advanced, básicamente supone llegar a velocidades de hasta 1Gbps en DL, y 500 Mbps en UL como exige la norma para poder ser clasificada en 4G.

Actualmente esta tecnología solo está desplegada en una pequeña fracción de Moscú.

A continuación un gráfico comparativo de las velocidades pico que se pueden alcanzar. Es destacable comentar que las velocidades que se pueden conseguir con LTE son equiparables a las que ofrece la fibra óptica.

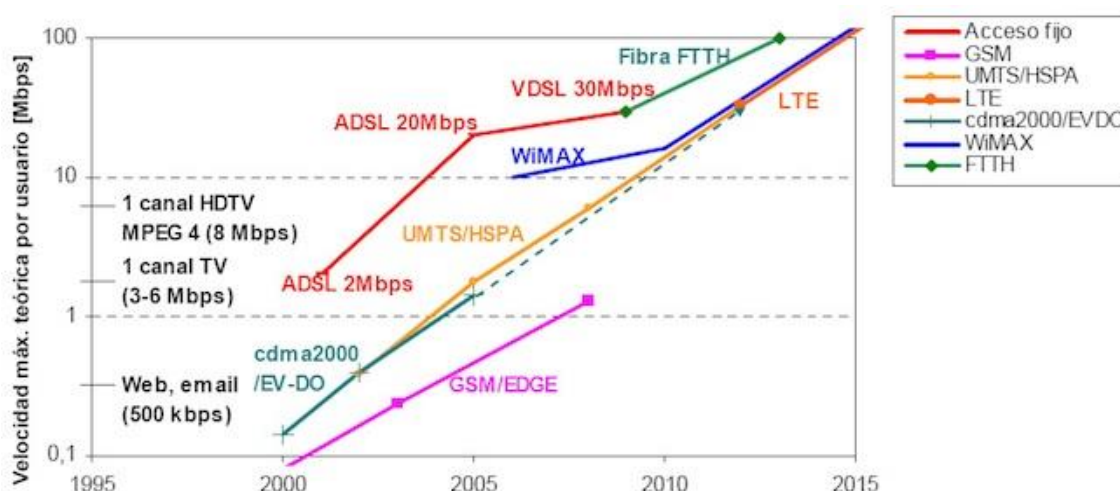


Figura 6: Gráfica comparativa de velocidades

OTROS

Por supuesto no son los únicos estándares de comunicaciones móviles que existen, en la siguiente tabla se puede apreciar que aparte del 3GPP hay otros organismos que también desarrollan sus propios estándares como el IEEE o el 3GPP2.



V · T · E		Cellular network standards
0G (radio telephones)		MTS · MTA * MTB * MTC · IMTS · MTD · AMTS · OLT · Autoradiopuhelin · B-Netz
1G	AMPS family	AMPS (TIA/EIA/IS-3, ANSI/TIA/EIA-553) · N-AMPS (TIA/EIA/IS-91) · TACS · ETACS
	Other	NMT · C-450 · Hicap · Mobitex · DataTAC
2G	GSM/3GPP family	GSM · CSD
	3GPP2 family	cdmaOne (TIA/EIA/IS-95 and ANSI-J-STD 008)
	AMPS family	D-AMPS (IS-54 and IS-136)
	Other	CDPD · IDEN · PDC · PHS
	GSM/3GPP family	HSCSD · GPRS · EDGE/EGPRS (UWC-136)
2G transitional (2.5G, 2.75G)	3GPP2 family	CDMA2000 1X (TIA/EIA/IS-2000) · 1X Advanced
	Other	WIDEN
3G (IMT-2000)	3GPP family	UMTS (UTRAN) · WCDMA-FDD · WCDMA-TDD · UTRA-TDD LCR (TD-SCDMA)
	3GPP2 family	CDMA2000 1xEV-DO Release 0 (TIA/IS-856)
3G transitional (3.5G, 3.75G, 3.9G)	3GPP family	HSPA · HSPA+ · HSDPA · LTE (E-UTRA)
	3GPP2 family	CDMA2000 1xEV-DO Revision A (TIA/EIA/IS-856-A)
		EV-DO Revision B (TIA/EIA/IS-856-B) · DO Advanced
	IEEE family	Mobile WiMAX (IEEE 802.16e) · Flash-OFDM · IEEE 802.20
4G (IMT Advanced)	3GPP family	LTE Advanced (E-UTRA)
	IEEE family	WiMAX-Advanced (IEEE 802.16m)

Figura 7: Tabla de estándares de comunicaciones móviles (6)

2.3.Android



Figura 8: Logo de Android

2.3.1. Descripción

Android es un sistema operativo pensado para dispositivos móviles al igual que iOS, Symbian y Blackberry OS, lo que lo diferencia es que está basado en Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma.

Ésta libertad le confiere:

- Popularidad entre fabricantes y desarrolladores ya que los costes para lanzar un teléfono o una aplicación son muy bajos.
- Seguridad para los usuarios, ya que al ser abierto permite detectar fallos más rápidamente y también para los fabricantes, ya que pueden adaptar mejor el sistema operativo a los terminales.



Ejecuta una máquina virtual similar a Java, llamada Dalvik, la cual permita ejecutar aplicaciones programadas en Java. En muchos casos se pueden importar librerías de Java (del código fuente) directamente sobre las aplicaciones del teléfono. Android asegura un cierto grado de portabilidad entre terminales de diferentes fabricantes.

2.3.2. Entorno

Existen múltiples entornos para programar Android pero se utilizará Eclipse que es el proporcionado por la web de desarrolladores de Android. (7)



Figura 9: Logo Eclipse

Eclipse permite programar en numerosos lenguajes, para desarrollar aplicaciones en Android es necesario el complemento ADT (Android Developer Tools), más el SDK. En la web de Android se suministra el kit completo. (8)



Figura 10: Logo ADT

El SDK de Android proporciona las librerías API y las herramientas de desarrollo necesarias para crear, testear y depurar aplicaciones para Android.

El paquete ADT incluye todo lo necesario para empezar a desarrollar aplicaciones, incluye los componentes esenciales del SDK de Android y una versión del Eclipse IDE (Integrated Development Integrated) incluyendo el ADT para agilizar el desarrollo de la aplicación Android.

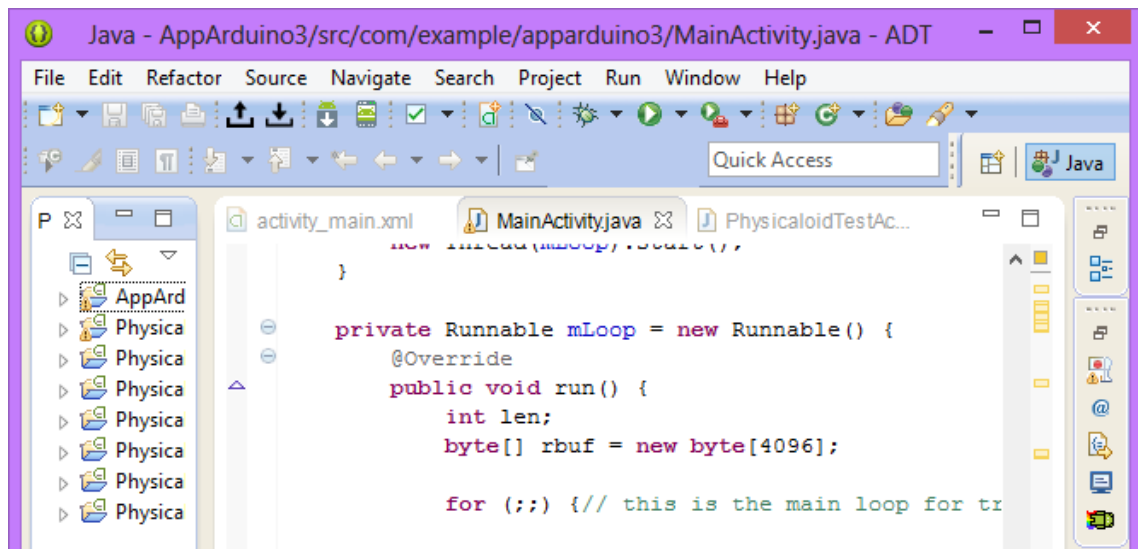


Figura 11: Interfaz Eclipse

Eclipse es un entorno de desarrollo para varios lenguajes, el manejo es similar en todos los lenguajes, pero en el caso de Android hay que añadir:

- El método para la depuración de errores se realiza mediante el LogCat. LogCat es el sistema de registro de Android que proporciona un mecanismo para recopilar y ver los mensajes de depuración de los sucesos del sistema. El dispositivo Android debe estar en modo depuración.

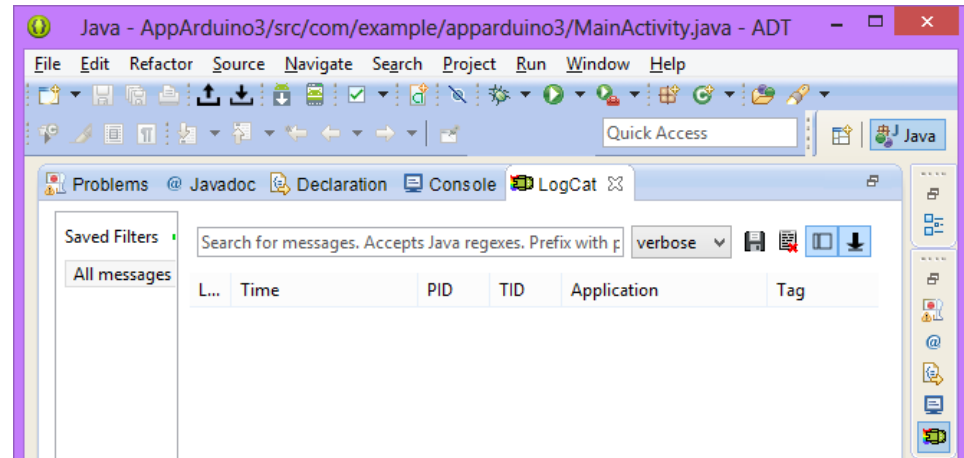


Figura 12: LogCat de Eclipse

- Para ejecutar la aplicación y probar su funcionamiento se puede hacer con:
 - El emulador incluido AVD (Android Virtual Device), se crea un dispositivo con las características que se deseen o tomando uno de los predefinidos.
 - Directamente en el dispositivo:
 - Automáticamente, teniendo el dispositivo conectado por cable al ordenador y en modo depuración, clicando en compilar se instalará automáticamente.



- Manualmente, copiando el .apk del directorio bin del workspace donde se haya creado el proyecto, mientras está en modo almacenamiento USB pegar en la tarjeta SD de la tableta, para encontrar .apk quitar modo almacenamiento USB y buscar con un explorador de archivos, como por ejemplo “Androexplorer”.

En la siguiente imagen se aprecia cómo deben disponerse las conexiones de cables, debe haber una conexión entre ordenador y tableta que permita cargar la aplicación, asimismo es necesaria para visualizar los eventos el registro de mensajes LogCat; además debe existir otra conexión entre tableta y Arduino para poder manejarlo.

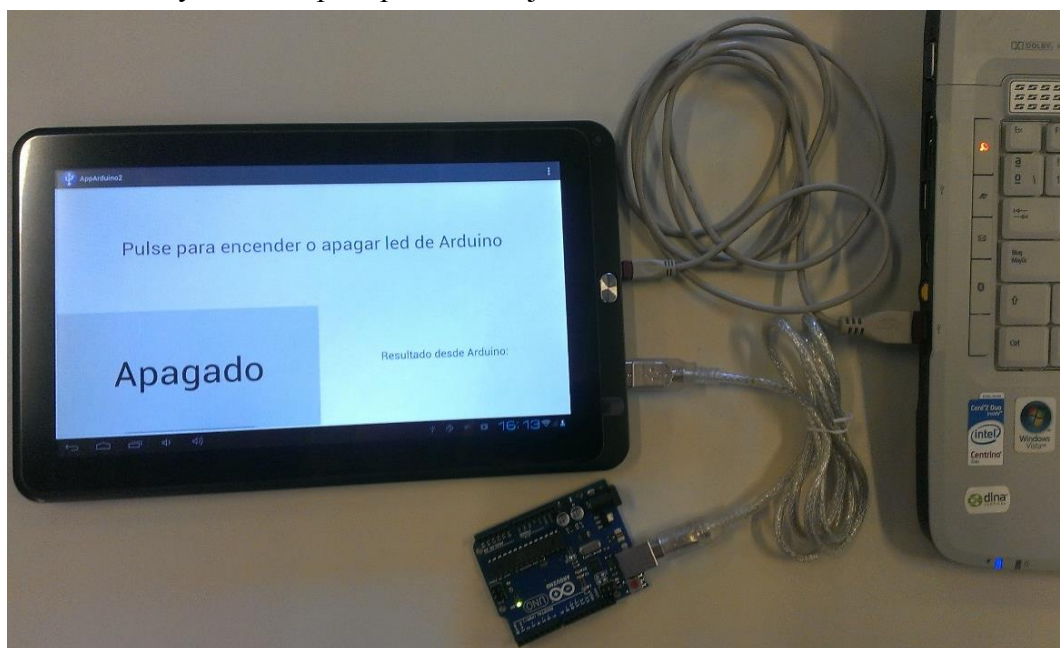


Figura 13: Imagen propia del proyecto en cuestión

2.3.3. Lenguaje

Para programar aplicaciones en Android se disponen de varios lenguajes pero normalmente Java es el más comúnmente utilizado.

Java es el lenguaje escogido ya que es el que ofrece la web de desarrolladores de Android y viene incorporado en Eclipse.

Estructura de un proyecto Android

Los proyectos de Android siguen siempre una estructura fija de carpetas que se deben mantener. Se puede observar fácilmente usando la vista de Package Explorer.

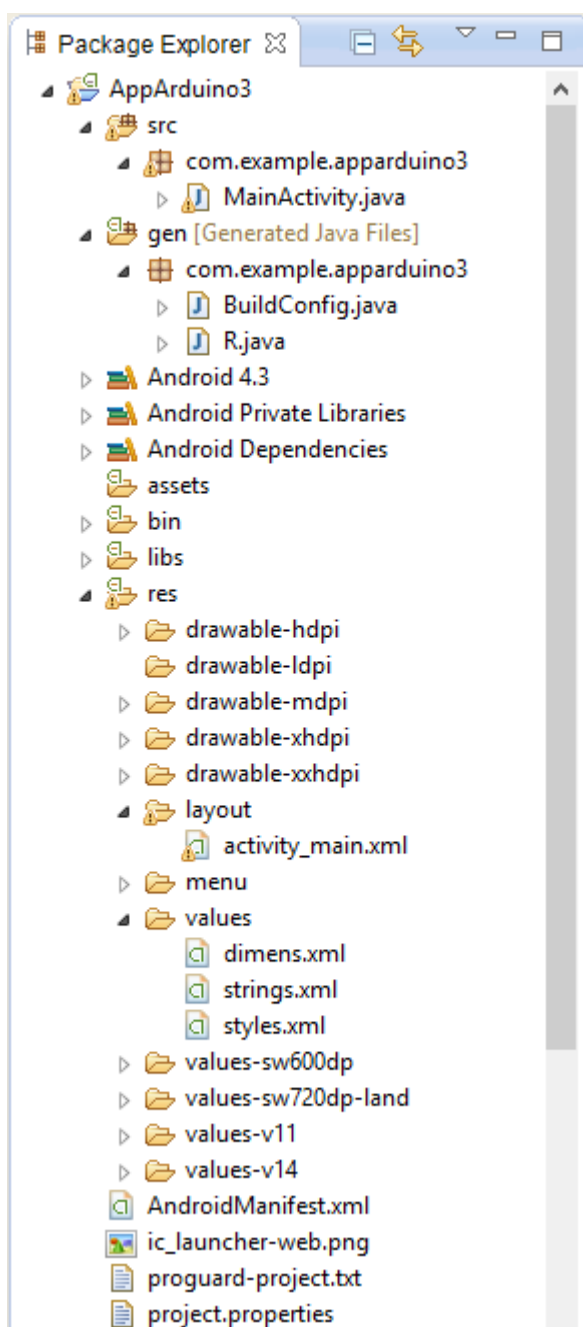


Figura 14: Estructura de carpetas Android

- La carpeta Android de fuentes (src): La primera carpeta que se encuentra es “src”. Contiene el código fuente organizado en paquetes. En esta carpeta es donde se colocan las clases java que se programan.
- La carpeta Android de archivos generados (gen): En ella se colocan los archivos que el compilador genera en sus repasos. Como el archivo de recursos R.
- La carpeta Android de recursos varios (assets): Sirve para almacenar recursos que pueda necesitar la aplicación como por ejemplo ficheros de música, zip, etc. Se puede acceder a ellos como si se tratara de un sistema de ficheros gracias a la clase del sistema AssetManager.





- La carpeta Android de recursos (res): Esta carpeta, junto con la carpeta de fuentes es la que más se utiliza. Contiene todos los recursos que necesita la aplicación como definición de pantallas, iconos, gráficos, cadenas de texto localizadas...
Todos los archivos que se colocan aquí son indexados por el compilador y se genera el fichero de recursos “R” que permite acceder a ellos de una forma rápida. Debido a la gran variedad de archivos que puede almacenar, está dividida en subcarpetas:
 - anim: ficheros xml de definición de animaciones.
 - color: ficheros xml de definición de colores.
 - drawable: ficheros bitmap (.png, .9.png, .jpg, .gif) o xml con contenidos que se “dibujarán”, es decir, iconos, imágenes, fondos, definición de botones...
 - layout: ficheros xml que definen la capa de interfaz de usuario.
 - menu: ficheros xml con la definición de los menús de la aplicación.
 - raw: ficheros binarios que no tienen cabida en el resto de carpetas.
 - values: ficheros xml de definición de valores simples como estilos, cadenas de texto para localización.
 - xml: archivos xml varios que pueden ser accedidos en tiempo de ejecución con el método `Resources.getXml()`.
- El manifiesto Android (AndroidManifest.xml): Todos los proyectos deben tener un archivo en la carpeta principal, de nombre AndroidManifest.xml en el que se detallan las características principales de la aplicación (Cada Activity se define aquí, los widgets, módulos principales, permisos necesarios que requiere la aplicación, el nombre de la app, el icono de la app...).

Componentes de una aplicación Android

En Android se disponen de los siguientes elementos básicos:

Activity: Las actividades (*activities*) representan el componente principal de la interfaz gráfica de una aplicación Android. Se puede pensar en una actividad como el elemento análogo a una ventana o pantalla en cualquier otro lenguaje visual.

View: Las vistas (*view*) son los componentes básicos con los que se construye la interfaz gráfica de la aplicación, análoga por ejemplo a los *controles* de Java. De inicio, Android pone a disposición una gran cantidad de controles básicos, como cuadros de texto, botones, listas desplegables o imágenes, aunque también existe la posibilidad de extender la funcionalidad de estos controles básicos o crear controles personalizados.

Service: Los servicios (*service*) son componentes sin interfaz gráfica que se ejecutan en segundo plano. En concepto, son similares a los servicios presentes en cualquier otro sistema operativo. Los servicios pueden realizar cualquier tipo de acciones, por ejemplo actualizar datos, lanzar notificaciones, o incluso mostrar elementos visuales (p.ej. actividades) si se necesita en algún momento la interacción con del usuario.

Content Provider: Un proveedor de contenidos (*content provider*) es el mecanismo que se ha definido en Android para compartir datos entre aplicaciones.

Broadcast Receiver: Un *broadcast receiver* es un componente destinado a detectar y reaccionar ante determinados mensajes o eventos globales generados por el sistema (por ejemplo: “Batería baja”, “SMS recibido”, “Tarjeta SD insertada”...) o por otras aplicaciones





(cualquier aplicación puede generar mensajes (intents, en terminología Android) broadcast, es decir, no dirigidos a una aplicación concreta sino a cualquiera que quiera escucharlo).

Widget: Los *widgets* son elementos visuales, normalmente interactivos, que pueden mostrarse en la pantalla principal (*home screen*) del dispositivo Android y recibir actualizaciones periódicas. Permiten mostrar información de la aplicación al usuario directamente sobre la pantalla principal.

Intent: Un *intent* es el elemento básico de comunicación entre los distintos componentes Android descritos anteriormente. Se pueden entender como los mensajes o peticiones que son enviados entre los distintos componentes de una aplicación o entre distintas aplicaciones. Mediante un *intent* se puede mostrar una actividad desde cualquier otra, iniciar un servicio, enviar un mensaje *broadcast*, iniciar otra aplicación, etc.

2.3.4. Código y librerías

La programación de la aplicación es convencional en cuanto a que requiere de un diseño de interfaz gráfica y una clase principal que lo gestione todo, pero es necesario realizar una labor adicional, y es que para que tableta y Arduino puedan hablarse, se debe implementar la comunicación serie.

Para manejar la comunicación serie por USB, se dispone de la API: “Android USB-Host API” que requiere Android 3.1 o superior, pero para simplificar el trabajo, en este proyecto se utiliza una librería particular “Physicaloid Library”, creada por Keisuke Suzuki, dicha librería maneja las clases de la API citada y proporciona una ayuda para establecer la comunicación serie al ofrecer en unos pocos métodos las acciones de la API necesarias para tal efecto.

También existe otra librería particular similar para establecer la comunicación serie “usb-serial-for-android” propiedad de Mike Wakerly. (Mik3y).

Las dos librerías son muy similares pero la razón de escoger la primera es que existe una aplicación de código abierto en Google store que utiliza dicha librería y funciona perfectamente, de esta manera se tiene un ejemplo práctico y funcional de cómo usar la librería.

I. Librería Physicaloid



Figura 15: Logo Physicaloid



Physicaloid Library es una Librería de Android para la comunicación con plataformas de creación de prototipos (Arduino, mbed).



Figura 16: Conexión USB serie

1) Características

- Proyecto de librería de Android Java.
- Comunicación USB Serial.
- Cargar firmware en Arduino.
- No es necesario ser ROOT.
- Compatible con Android 3.1 o superior.
- Compatible con protocolos USB Serial: CDC-ACM, FTDI, Silicon Labs CP210x.
- Compatible con protocolos de carga de firmware: STK500, STK500V2.
- Código abierto (Licencia Apache 2.0).

2) Código de ejemplo

Escribir datos serie a Arduino

```
Physicaloid mPhysicaloid = new Physicaloid(this);
if(mPhysicaloid.open()) {
    byte[] buf = "moemoe".getBytes();
    mPhysicaloid.write(buf, buf.length);
    mPhysicaloid.close();
}
```

Leer datos serie desde Arduino

```
Physicaloid mPhysicaloid = new Physicaloid(this);
TextView TextView1 = (TextView) findViewById(R.id.TextView1); // Android TextView

if(mPhysicaloid.open()) {
    byte[] buf = new byte[256];

    mPhysicaloid.read(buf, buf.length);
}
```




```
String str = new String(buf);
TextView1.append(str);

mPhysicaloid.close();
}
```

3) Cómo usarla

1. File -> import y seleccionar el directorio PhysicaloidLibrary.
2. Botón derecho en el proyecto -> Propierties -> Android -> click en botón “Add” de la sección Library -> seleccionar PhysicaloidLibrary.

El proyecto contiene además de la librería, un tutorial/ejemplo de cómo usar cada característica de la librería.

4) Licencia

La librería Physicaloid está publicada bajo licencia Apache, versión 2.0.²

5) Descargas

El proyecto se encuentra en GitHub (9) donde se puede descargar, y en la web oficial (En Japonés) (10) se halla toda la documentación completa.

II. Librería usb-serial-for-android

Esta librería se basa en el API de Android USB Host al igual que la anterior, se ha probado en los componentes del proyecto y funciona, tiene además funciones muy parecidas a la anterior, tales como leer, escribir o abrir. Pero no se ha escogido por la razón dicha antes, Physicaloid tiene una aplicación de código abierto que la usa y se decidió empezar por esa ya que se tenía el respaldo de algo que funcionaba correctamente.

1) Características

- Librería para comunicación serial con Arduino
- Compatible con Android 3.1 o superior.
- No es necesario root, ADK o drivers de kernel específicos.
- Todos los drivers están implementados en java.
- Provee funciones básicas para usar el puerto serie
- Compatible con FT232R, CDC/ACM (ej. Arduino UNO) y otros posibles chips.
- Compatible con amplia gama de teléfonos y tabletas Android.

² La licencia Apache es una licencia de software libre, no exige que las obras derivadas (versiones modificadas) del software se distribuyan usando la misma licencia, ni siquiera que se tengan que distribuir como software libre/open source. La Licencia Apache sólo exige que se mantenga una noticia que informe a los receptores que en la distribución se ha usado código con la Licencia Apache.





2) Cómo usarla

1. Descargar [usb-serial-for-android-v010.jar](#)
2. Copiar el fichero jar en el directorio “libs/” del proyecto Android
3. Copiar [device_filter.xml](#) en el directorio “res/xml/” del proyecto Android
4. Configurar el fichero “AndroidManifest.xml” del proyecto para notificar a la aplicación cuando el dispositivo se conecte.

3) Código de ejemplo

Configuración del manifiesto

```
<activity
    android:name="..."
    ...>

<intent-filter>

    <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />

</intent-filter>

<meta-data

    android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"

    android:resource="@xml/device_filter" />

</activity>
```

Ejemplo de uso

```
// Get UsbManager from Android.

UsbManager manager = (UsbManager) getSystemService(Context.USB_SERVICE);

// Find the first available driver.

UsbSerialDriver driver = UsbSerialProber.acquire(manager);

if (driver != null) {

    driver.open();

    try {
```





```

driver.setBaudRate(115200);

byte buffer[] = new byte[16];

int numBytesRead = driver.read(buffer, 1000);

Log.d(TAG, "Read " + numBytesRead + " bytes.");
} catch (IOException e) {

    // Deal with error.

} finally {

    driver.close();

}
}

```

4) Licencia

Esta librería está publicada bajo licencia LGPL Versión 2.1

5) Descargas

El proyecto se puede encontrar en GitHub (11) y también en Google group (12).

III. API Host USB

Esta librería (13) proporciona métodos que permiten establecer una comunicación serie entre dispositivos Android y otros dispositivos que se conecten por USB.

Cuando el dispositivo Android está en modo USB Host, actúa como host USB, alimenta el bus, y enumera los dispositivos USB conectados. El modo USB host es compatible con Android 3.1 y superiores.

En la siguiente tabla se describen las APIs USB Host del paquete [android.hardware.usb](#)

Class	Descripción
UsbManager	Permite enumerar y comunicarse con los dispositivos USB conectados.
UsbDevice	Representa un dispositivo USB conectado y contiene métodos para acceder a su información de identificación, interfaces, y endpoints.



UsbInterface	Representa una interfaz de un dispositivo USB, que define un conjunto de funcionalidades para el dispositivo. Un dispositivo que puede tener una o más interfaces con las que comunicarse.
UsbEndpoint	Representa una interfaz endpoint, que es un canal de comunicación para esta interfaz. Una interfaz puede tener uno o más endpoints, y por lo general tiene extremos de entrada y salida para la comunicación de doble sentido con el dispositivo.
UsbDeviceConnection	Representa una conexión con el dispositivo, que transfiere los datos en los endpoints. Esta clase permite enviar datos de un lado a otro síncronamente o asíncronamente.
UsbRequest	Representa una solicitud asíncrona para comunicarse con un dispositivo a través de UsbDeviceConnection.
UsbConstants	Define constantes USB que se corresponden con las definiciones en linux/usb/ch9.h del kernel de Linux.

En la mayoría de situaciones, es necesario utilizar todas estas clases (UsbRequest sólo es necesario para la comunicación asíncrona) para la comunicación con un dispositivo USB. En general, se obtiene el UsbManager para recuperar el UsbDevice deseado. Cuando se tiene el dispositivo, es necesario encontrar el UsbInterface apropiado y el UsbEndpoint de esa interfaz para comunicarse. Una vez obtenido el endpoint correcto, abrir UsbDeviceConnection para comunicarse con el dispositivo USB.

2.4.Arduino



Figura 17: Logo Arduino

2.4.1. Descripción

Es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware libres diseñada para ser barata y fácil de usar. (14)



- ¿Qué es Arduino realmente?

Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software.

Las placas pueden ser hechas a mano o compradas montadas de fábrica, el software puede ser descargado de forma gratuita. Los ficheros de diseño de referencia (CAD) están disponibles bajo una licencia abierta.

Se ha buscado desde el principio desarrollar el sistema con una plataforma que fuese de código abierto. Arduino tiene todos los ingredientes para ser apetecible para alguien que busque profundizar sus conocimientos en el mundo de la electrónica. Es por eso que existe una gran comunidad de artistas, diseñadores y aficionados creando proyectos constantemente con este microcontrolador.

Arduino Uno



Figura 18: Placa Arduino UNO

- Ventajas

Existen muchos otros microcontroladores y plataformas con microcontroladores disponibles para la computación física. Parallax Basic Stamp, BX-24 de Netmedia, Phidgets, Handyboard del MIT, y muchos otros ofrecen funcionalidades similares. Todas estas herramientas organizan el complicado trabajo de programar un microcontrolador en paquetes fáciles de usar. Arduino, además de simplificar el proceso de trabajar con microcontroladores, ofrece algunas ventajas respecto a otros sistemas:





➤ Asequible

Las placas Arduino son más asequibles comparadas con otras plataformas de microcontroladores. La versión más cara de un módulo de Arduino puede ser montada a mano, e incluso ya montada cuesta bastante menos de 60€.

➤ Multi – Plataforma

El software de Arduino funciona en los sistemas operativos Windows, Macintosh OSX y Linux. La mayoría de los entornos para microcontroladores están limitados a Windows.

➤ Entorno de programación simple y directo

El entorno de programación de Arduino es fácil de usar para principiantes y lo suficientemente flexible para los usuarios avanzados. Pensando en los profesores, Arduino está basado en el entorno de programación de Processing con lo que el estudiante que aprenda a programar en este entorno se sentirá familiarizado con el entorno de desarrollo Arduino.

➤ Software ampliable y de código abierto

El software Arduino se publica bajo una licencia libre y preparado para ser ampliado por programadores experimentados. El lenguaje puede ampliarse a través de librerías de C++, y si se está interesado en profundizar en los detalles técnicos, se puede dar el salto a la programación en el lenguaje AVR C en el que está basado. De igual modo se puede añadir directamente código en AVR C en tus programas si así lo deseas.

➤ Hardware ampliable y de Código abierto

Arduino está basado en los microcontroladores ATMEGA168, ATMEGA328 y ATMEGA1280. Los planos de los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores de circuitos con experiencia pueden hacer su propia versión del módulo, ampliándolo u optimizándolo. Incluso usuarios relativamente inexpertos pueden construir la versión para placa de desarrollo para entender cómo funciona y ahorrar algo de dinero.

• Especificaciones técnicas

El modelo utilizado en concreto es Arduino UNO rev3.

Microcontrolador	ATmega328
Voltaje de funcionamiento	5V
Alimentación(recomendada)	7-12V





Voltaje máximo de entrada (no recomendado)	20V
Pines digitales I/O	14 (de los cuales 6 dan salida PWM)
Pines de entrada analógica	6
Corriente DC por I/O Pin	40 mA
Corriente DC para el pin 3.3V	50 mA
Memoria Flash	32 KB (ATmega328) 0.5 KB usados por el bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidad de reloj	16 MHz

Dispone de un procesador ATMEL ATmega328 a 16 MHz. Memoria Flash de 32 KB y 1 KB de memoria EEPROM (memoria que no se borra al cortar el suministro eléctrico). En cuanto a los pines, operan a 5 V. Tiene 14 digitales de los cuales 6 pueden actuar como salida de tipo PWM. Por último, tenemos los pines 0(RX) y 1(TX) (Receive y Transmit respectivamente) que permiten una comunicación TTL en serie.

- Tipos de Arduino

Existen diferentes modelos de Arduino para ajustarse a las necesidades de cada proyecto. A la hora de elegir lo normal es fijarse en la cantidad de entradas y salidas que tiene, especialmente las analógicas, dado que son las que habitualmente restringen el proyecto.

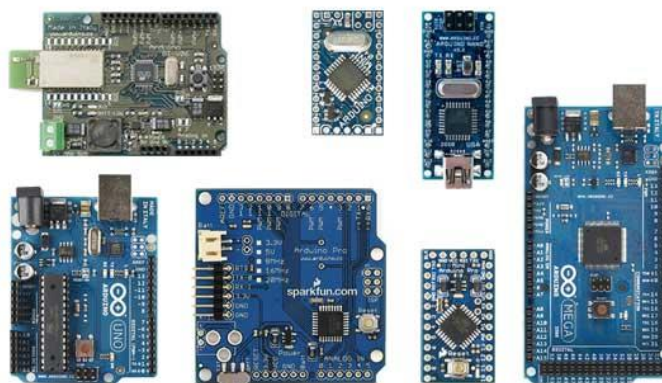


Figura 19: Modelos de Arduino

- Accesorios/Shields/módulos

Un shield o módulo, es una placa impresa que se puede conectar en la parte superior de la placa Arduino para ampliar sus capacidades ofreciendo nuevos servicios, por ejemplo: puertos de red, tarjetas bluetooth, USB, entre otros.

Los módulos permiten añadir funcionalidades al Arduino al instante. Se pueden apilar unos encima de otros y aprovechar todas las funcionalidades que ofrecen.



Los shields carecen de la capacidad de procesamiento principal para ejecutar el código escrito en sketches o bocetos.

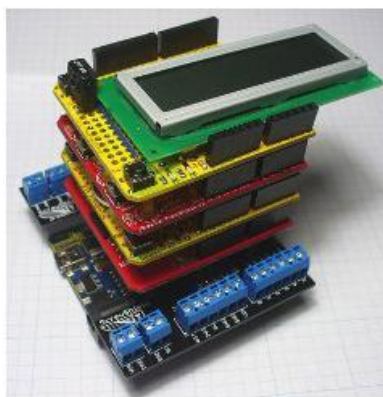


Figura 20: Shields apilados

2.4.2. Entorno de desarrollo para Arduino

El entorno de Desarrollo Arduino está constituido por un editor de texto para escribir el código, un área de mensajes, una consola de texto, una barra de herramientas con botones para las funciones comunes, y una serie de menús. Permite la conexión con el hardware de Arduino para cargar los programas y comunicarse con ellos.

Arduino utiliza para escribir el software lo que denomina "sketch" (*programa*). Estos programas son escritos en el editor de texto. Existe la posibilidad de cortar/pegar y buscar/remplazar texto. En el área de mensajes se muestra información mientras se cargan los programas y también muestra errores. La consola muestra el texto de salida para el entorno de Arduino incluyendo los mensajes de error completos y otras informaciones. La barra de herramientas permite verificar el proceso de carga, creación, apertura y guardado de programas, y la monitorización serie.

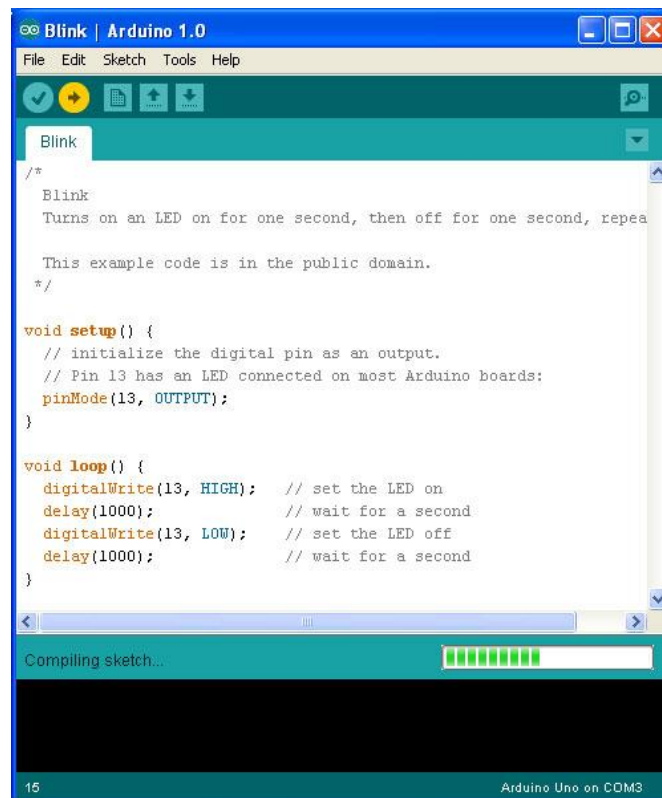


Figura 21: Entorno Arduino

2.4.3. Lenguaje

La plataforma Arduino se programa mediante el uso de un lenguaje propio (15) basado en el popular lenguaje de programación de alto nivel Processing. Sin embargo, es posible utilizar otros lenguajes de programación y aplicaciones populares en Arduino, esto es posible debido a que Arduino se comunica mediante la transmisión de datos en formato serie que es algo que la mayoría de los lenguajes anteriormente citados soportan.

Por lo general la estructura de un programa en Arduino (extensión .ino) consta de tres partes:

- Declaración de variables: Inicializar o dar valores a las variables (expresiones que almacenan valores).
- Función de inicialización (método void setup()): Configuración inicial del chip, inicialización de variables, modos de pin, empezar a utilizar librerías. Se ejecuta al inicio una vez (o al reiniciar el Arduino en el botón físico que se encuentra en la placa).
- Función de programa (método void loop()) que se ejecutará infinitamente): Aquí va la funcionalidad de lo que se está implementando. Puede ser una máquina de estados simple o simples operaciones que ejecutamos continuamente, el tiempo del bucle, varía según el número de instrucciones que contenga. Y se puede conocer y controlar con las funciones de temporización (millis(), delay(), etc).





2.4.4. Código

El sketch que se va a utilizar para probar la comunicación entre la plataforma Arduino y el dispositivo Android es un código muy simple, con el cual se podrá encender y apagar el LED del pin 13 que tiene el Arduino UNO, al recibir un “1” se encenderá y contestará con la cadena de caracteres “ON “, y al recibir un “0” se apagará y contestará con “OFF “.

```
int led = 13; // El Pin 13 tiene un LED conectado en Arduino
int incomingByte = 0; // Para los datos en serie entrantes

// La rutina setup se ejecuta una vez al pulsar reset:
void setup() {
    pinMode(led, OUTPUT); // Se inicializa el pin digital como salida.
    Serial.begin(9600); // se abre el puerto serie, fija la velocidad de
    datos a 9600 bps.
}

// La rutina loop se ejecuta una y otra vez para siempre:
void loop() {
    // Envía datos solo cuando recibe datos:
    if (Serial.available() > 0) { //Devuelve el número de bytes
    (caracteres) disponibles para ser leídos por el puerto serie.
    // Se refiere a datos ya recibidos y disponibles en el buffer de recepción
    del puerto (que tiene una capacidad de 128 bytes).
        // Lee el byte entrante que llega por el puerto serie:
        incomingByte = Serial.read();
        if (incomingByte == '1'){
            digitalWrite(led, HIGH); //Enciende LED (HIGH es el nivel de
    voltaje)

            //Envía por puerto serie el String: "ON "
            Serial.print("ON ");
        }
        if (incomingByte == '0'){
            digitalWrite(led, LOW); //Apaga LED (LOW es el nivel de
    voltaje)

            Serial.print("OFF "); //Envía por puerto serie el String: "OFF "
        }
        delay(10);
    }
}
```



2.5. Módulo 3G

2.5.1. Descripción

- ¿Qué es?

El Módulo 3G con el cuál se va a trabajar, es un módulo/shield compatible con la plataforma Arduino, permite el uso de las facilidades que proporciona la tecnología de comunicaciones móviles. (16)



Figura 22: Módulo 3G

La placa contiene un módulo SIM5218E que permite utilizar los servicios de comunicaciones móviles. Los sistemas soportados son UMTS/HSPA 900/1900/2100 MHz y GSM/GPRS/EDGE 850/900/1800/1900 MHz, es decir que posibilita la conexión 2G y 3G. Al igual que un teléfono móvil convencional, se pueden realizar llamadas, enviar SMS, navegar por internet entre otras cosas.



Figura 23: Módulo SIM5218E

Además, el Módulo 3G cuenta con GPS (Global Positioning System), cámara de video, micrófono, altavoz, altavoz manos libres y auriculares, ranura SD con tarjeta SD para guardar directamente la información o reproducirla (como música) desde ahí.

También se puede utilizar como módem 3G con velocidades de hasta 7.2Mbps en enlace descendiente y hasta 5.5Mbps en enlace ascendiente conectándolo directamente al ordenador por USB

Esta versión del Módulo 3G está especialmente orientada a trabajar con protocolos de internet que hace más sencillo la subida de información a la nube. Se puede navegar por HTTP HTTPS, descargar y subir contenidos. De igual manera los protocolos FTP y FTPS están disponibles, muy útil a la hora de manejar ficheros. Y se pueden enviar y recibir e-mails utilizando clientes de correo SMTP y POP3.

El GPS permite utilizar los servicios de geo localización incluso en interiores ya que puede trabajar en los modos A-GPS y S-GPS, consiguiendo una localización más precisa.

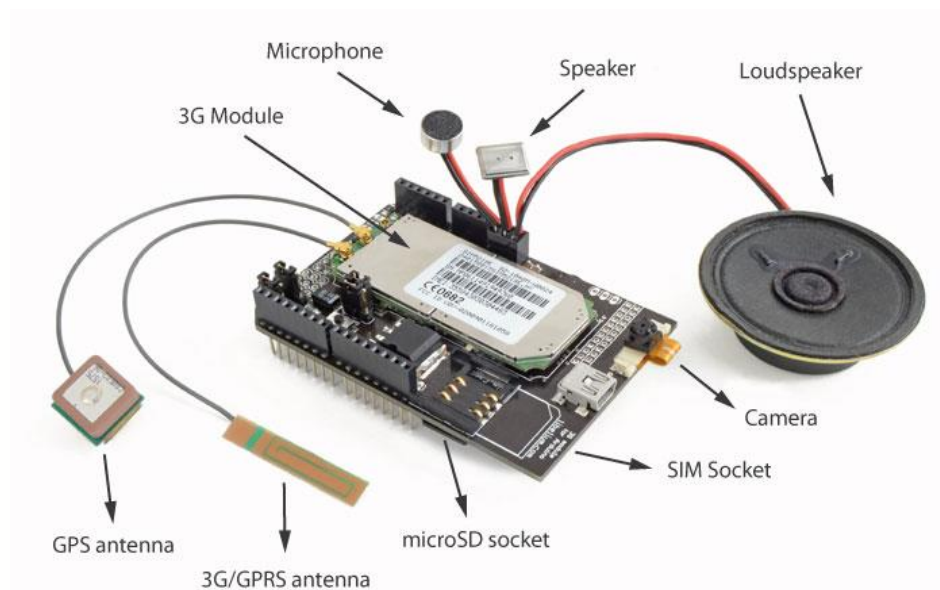


Figura 24: Módulo 3G con sus accesorios

El paquete adquirido en la tienda (17) incluye todos los accesorios previamente mostrados.

- **Características**

- Compatibilidad con tecnologías WCDMA y HSPA
- GPS integrado con modos Assisted A-GPS y Supported S-GPS
- Cámara (640x480) para vídeos y fotos
- Kit de audio que incluye micrófono, altavoz, altavoz manos libres y auriculares
- Sistema de ficheros SD ampliable hasta 32GB
- Funciona como un módem 3G standard en Linux/Windows/MacOS (hasta 7,2Mbps de bajada y hasta 5,5Mbps de subida)
- Comunicación directa con los servidores web mediante HTTP/HTTPS
- Carga y descarga de archivos directamente por FTP/FTPS
- Envío y recepción de correos por POP3/SMTP
- Reproducción de archivos de audio comprimidos



- Diagrama de conexiones del Módulo 3G

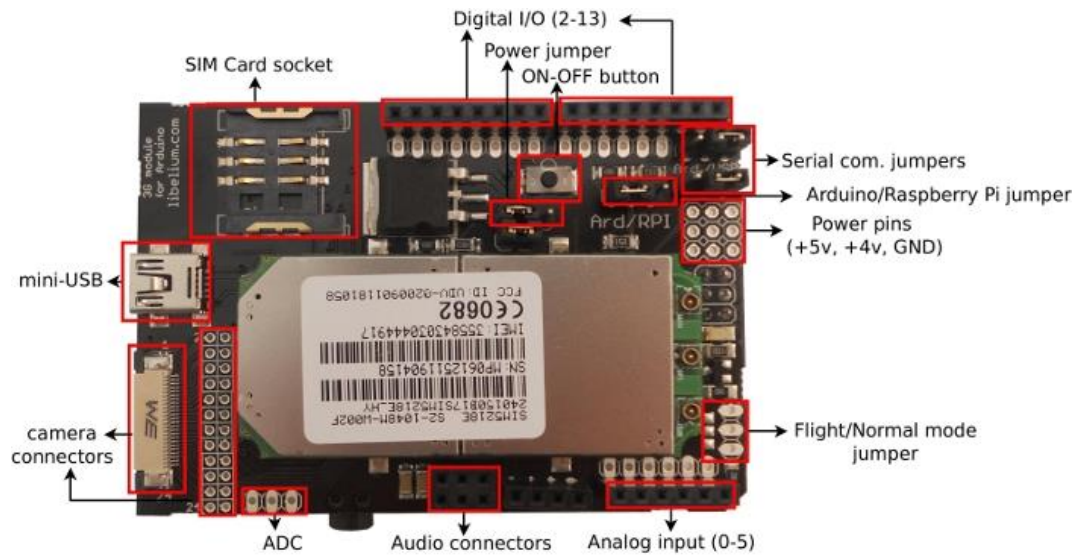


Figura 25: Diagrama de conexiones del Módulo 3G (Por delante)

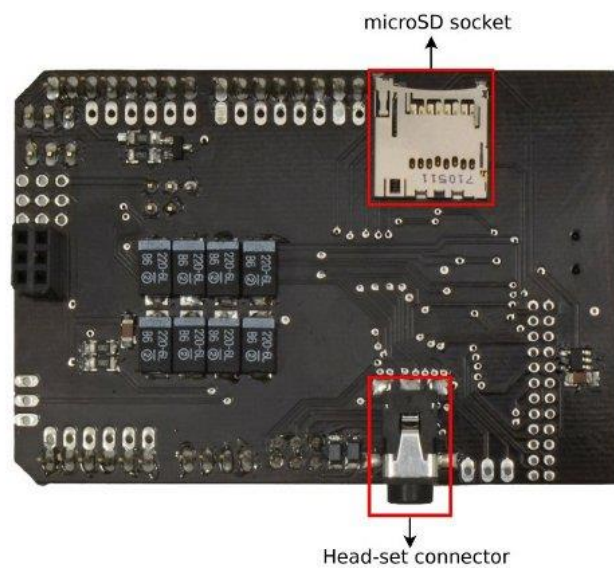


Figura 26: Diagrama de conexiones del Módulo 3G (Por detrás)



- Especificaciones técnicas

<p>General features</p> <ul style="list-style-type: none">• SIM5218A: Tri-Band UMTS/HSPA850/1900/2100 MHzSIM5218E :Tri-Band UMTS/HSPA 900/1900/2100MHzSIM5218C: Single-Band UMTS 2100MHzSIM5218J: Tri-Band UMTS/HSPA800/1900/2100 MHz• SIM5218A/E/C/J: Quad-Band GSM/GPRS/EDGE 850/900/1800/1900• Video call with Camera sensor interface• WCDMA Diversity• HSDPA 7.2Mbps• HSUPA: 5.76Mbps• GPRS multi-slot class 12• EDGE multi-slot Class 12• Output power- UMTS 850/1900/1900/2100: 0.25W- GSM850/GSM900: 2W- DCS1800/PCS1900: 1W• Control Via AT Commands• Supply voltage range: 3.4 ~ 4.2V• Dimension: 58 * 26 * 4.5mm• Weight: 15g• Normal operation temperature: -20°C to +70°C• Restricted operation temperature: -30°C to -20°C; +70°C to +80°C• Storage temperature: -40°C to +85°C <p>Specifications for Data</p> <ul style="list-style-type: none">• HSDPA - Max. 7.2Mbps (DL), - UE Category: 4, 6, 8, 12• HSUPA - Max.5.76Mbps(UL)• HSUPA+HSDPA- Max. 2.0Mbps (UL) +3.6Mbps (DL)• UMTS- Max.384Kbps (DL), Max.384Kbps (UL)• EDGE Class- Max. 236.8kbps(DL),Max.118Kbps(UL)• GPRS- Max. 85.6kbps(DL), Max.42.8kbps(UL)• CSD- GSM Data rate 14.4Kbps- UMTS date rate 57.6kbps <p>Specifications for A-GPS/GPS</p> <ul style="list-style-type: none">• A-GPS mode: MS-Based, MS-Assisted• Stand-alone GPS <p>Specifications for Network Identity and Time zone (NITZ)</p> <ul style="list-style-type: none">• support on GSM and WCDMA	<p>Specifications for video call:</p> <ul style="list-style-type: none">• DTMF on H245• Support standard WCDMA 64kbps (CS) Video call <p>Specification for SMS:</p> <ul style="list-style-type: none">• Point to point MO and MT• Text and PDU mode <p>Specification for Voice:</p> <ul style="list-style-type: none">• Voice Codec HR, FR, EFR, AMR• DTMF <p>Other features:</p> <ul style="list-style-type: none">• USB Driver for Microsoft Windows 2000/XP/Nista/Windows mobile/ Windows CE• USB Driver for Linux 2.6.16• Firmware update via USB• MMS• TCP/IP• FTP/FTPS/HTTP/HTTPS/SMTP/POP3• FOTA <p>Interfaces</p> <ul style="list-style-type: none">• Embedded SIM card (option)• USB2.0 high speed• UART• USIM• Audio: 2 Microphone, 3 speakers• Camera interface• SD card interface• I2C interface• GPIO <p>Certifications</p> <p>SIM5218A • CE</p> <ul style="list-style-type: none">• FCC• ROHS <p>SIM5218E • CE</p> <ul style="list-style-type: none">• FCC• ICACA• ROHS <p>SIM5218J • Telec</p> <ul style="list-style-type: none">• JATE <p>Carrier Approvals</p> <p>SIM5218E • NCC</p> <ul style="list-style-type: none">• TIM
---	--





2.5.2. Comandos AT (ATtention)

Los comandos AT son un lenguaje de comunicación compuesto de instrucciones cuya finalidad es la configuración de terminales módem.

La finalidad principal de los comandos AT es la comunicación con módems pero la telefonía móvil GSM también ha adoptado como estándar este lenguaje para poder comunicarse con sus terminales. De esta forma, todos los teléfonos móviles GSM poseen un juego de comandos AT específico que sirve de interfaz para configurar y proporcionar instrucciones a los terminales, permiten acciones tales como realizar llamadas de datos o de voz, leer y escribir en la agenda de contactos y enviar mensajes SMS, además de muchas otras opciones de configuración del terminal.

2.6. Comunicación serie

La comunicación serie o serial (18), se basa en la transmisión secuencial de la información, bit a bit, uno tras otro y a un ritmo acordado entre el emisor y el receptor.

Existen distintos tipos de interfaces que permitan la comunicación serie, el puerto USB (Universal Serial Bus) que permite mayor versatilidad en la conexión de múltiples dispositivos, el tradicional puerto serie que atiende a los estándares RS-232 (Recommended Standard 232), RS-485, I2C, Ethernet, PS2, etc (19).

Hay dos tipos de comunicación serie, síncrona y asíncrona, su principal diferencia es la forma en la que la información de sincronización (información de configuración de cómo van a ser enviados los datos) es transmitida del emisor al receptor, en la síncrona es necesario una señal de reloj (clock) para sincronizar los dispositivos y por lo tanto se necesita una línea física adicional dedicada al clock; en la asíncrona (UART, USB) no se usa una señal de reloj, se transmite la información por un única línea de datos y la información de sincronización puede ir incluida en los propios datos y ser extraída de éstos en recepción, o puede ser pactada por los comunicantes de antemano.

Arduino es capaz de comunicarse con otros componentes (memorias, convertidores, etc) o con otros dispositivos externos (ordenador, otros microcontroladores), esto es posible ya que el microcontrolador de Arduino posee un puerto serie conocido como UART (Universal Asynchronous Receiver and Transmitter) que permite la comunicación serial asíncrona full-duplex (transmisión en ambos sentidos simultáneamente) y requiere dos líneas de datos.

Para comunicarse con los ordenadores a través de los puertos USB y enviar y recibir datos de forma serial se requiere un dispositivo “traductor”, un convertidor USB Serial, Arduino lleva integrado el FT232R. Esto permite que la placa sea reconocida en el ordenador como un dispositivo conectado a un puerto COM aun siendo la conexión física por USB.

En Arduino la comunicación serie es asíncrona, por lo tanto, además de realizar la conexión física entre los dispositivos, es necesario pactar previamente la información de sincronización con los siguientes parámetros:





- Velocidad de transmisión (Baud rate): Número de símbolos por segundo que se transmiten, medido en baudios. Velocidades permitidas en Arduino 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 y 115200.
- Bits de datos (Data bits): Cantidad de bits de la transmisión. El tamaño del paquete puede ser de 5, 6, 7, u 8 bits.
- Bits de parada (Stop bit): Indica el fin de la transmisión de un solo paquete, puede valer 1, 1'5, o 2 bits.
- Paridad (Parity check): Para la detección de errores en la transmisión, hay cinco tipos: ninguna, par, impar, marcada y espaciada



3. Fases de desarrollo

Para llegar a obtener el control del Módulo 3G, se divide el desarrollo en dos etapas con el propósito de hacerlo más sencillo, en la primera se pretende realizar una toma de contacto con la plataforma Arduino, conseguir establecer una comunicación y probarla mediante una aplicación de demostración; en la segunda fase se busca la manera en la que se pueda entablar una comunicación con el Módulo 3G, para finalmente demostrar el control del Módulo 3G con una aplicación ejemplo.

3.1. Primera fase. Control de Arduino.

3.1.1. Requisitos

- ✓ Dispositivo Android (móvil o tableta) con puerto USB o micro USB en este último caso además sería necesario un adaptador OTG. [Consultar Apéndice I: B. Cable USB.](#)
- ✓ Android 3.1
- ✓ Cable USB tipo A – tipo B
- ✓ Arduino UNO

Los medios utilizados son:

- Tableta PROTAB2XXL (20) con puerto USB tipo A
- Cable USB tipo A – tipo B (No OTG)
- Android 4.0.3
- Arduino UNO rev3

3.1.2. Instrucciones

En esta fase se indican los pasos a seguir para establecer una comunicación entre un dispositivo Android y la plataforma Arduino, y posteriormente desarrollar aplicaciones Android que manejen Arduino:

1. Comprobar que el dispositivo Android (la tableta en este caso) permite la conexión mediante su puesta en modo USB host. [Consultar Apéndice I: A. Cómo saber si el dispositivo Android tiene soporte USB Host mode.](#)
2. Instalar el SDK y la plataforma de desarrollo Eclipse. [Consultar Apéndice F: Instalación de Android en Windows.](#)
3. Configurar drivers ADB para el reconocimiento del dispositivo Android en PC, sólo es necesario si Windows no lo reconoce en el Administrador de dispositivos. [Consultar Apéndice D: Instalación de ADB drivers.](#)
4. Instalar el entorno de programación Arduino, codificar el Sketch y subirlo a Arduino. [Consultar Apéndice E: Guía de instalación de Arduino en Windows.](#) El código del Sketch que se va a utilizar se ha mencionado anteriormente. [Ver apartado 2.4.4. Código.](#)
5. Para comprobar que el sketch funciona correctamente antes de codificar la aplicación en Android, se puede hacer de dos formas: teniendo la plataforma Arduino conectada al ordenador y utilizando el monitor serie del IDE Arduino para enviar y recibir; o, teniendo Arduino conectado a la tableta, instalar en Android alguna aplicación





disponible en Google Play que funcione de monitor serie, como por ejemplo “Android USB Serial Monitor Lite”.

6. En Eclipse, desarrollar el código para controlar Arduino con ayuda de una librería que maneje la comunicación serie. [Ver apartado 2.3.4. Código y librerías.](#)
7. Instalar y ejecutarlo en el dispositivo Android. [Ver apartado 2.3.2 Entorno.](#)

3.1.3. Resultado final

El resultado final es el siguiente montaje:

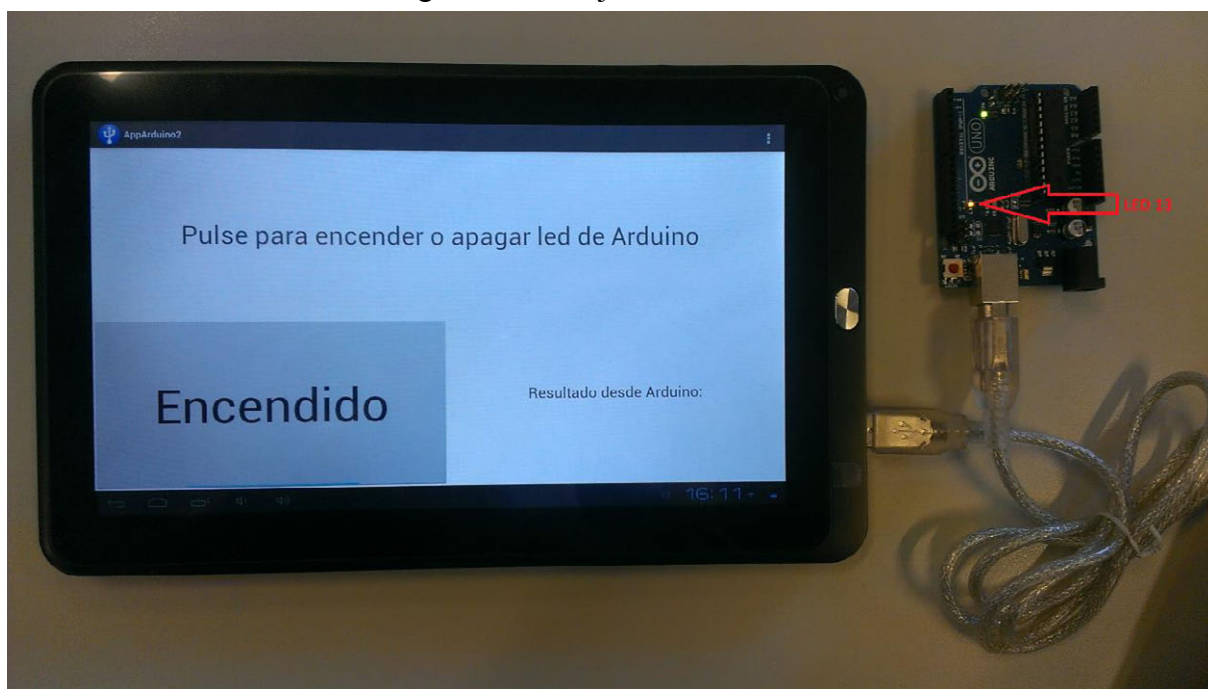


Figura 27: Tableta con AppArduino conectada a Arduino

La aplicación en Android llamada AppArduino controla el encendido y apagado de la luz del LED 13 de la plataforma Arduino.

3.2. Segunda fase. Control de Módulo 3G.

3.2.1. Requisitos

- ✓ Los requisitos anteriormente solicitados para el control de Arduino son necesarios también para esta fase, puesto que la comunicación serie conseguida con Arduino va a servir de base para establecer comunicación con el Módulo 3G
- ✓ Paquete Módulo 3G (placa, antena 2G/3G, micrófono, altavoz, cables conectores)
- ✓ Tarjeta SIM

Los medios utilizados son:

- ✓ Tableta PROTAB2XXL (20) con puerto USB tipo A
- ✓ Cable USB tipo A – tipo B (No OTG)
- ✓ Android 4.0.3
- ✓ Arduino UNO rev3



3.2.2. Instrucciones

En esta fase se describen los pasos para controlar el Módulo 3G desde el dispositivo Android, se debe haber realizado previamente el control de Arduino siguiendo los pasos del apartado anterior “Control de Arduino” ya que será necesario para lograr el objetivo.

1. Instalar los controladores del Módulo 3G en el ordenador para probarlo desde ahí. [Consultar Apéndice II. G. Instalación del Módulo 3G para Windows 8.](#)
2. Comprobar el correcto funcionamiento del Módulo 3G mediante un programa de ordenador que pueda comunicarse utilizando comandos AT y así tener un sistema fiable de referencia donde poder realizar pruebas, por si acaso en el desarrollo de la aplicación Android la ejecución de los comandos AT dan fallos. [Consultar Apéndice II. H Introducción a HyperTerminal.](#)
3. Realizar los ajustes necesarios en la placa Arduino y en la placa 3G para conseguir la comunicación entre el Módulo 3G y Android. [Consultar Apéndice II. I. Configuración para comunicación entre el Módulo3G y Android.](#)
4. Conectar lo elementos de red (SIM y antena) y audio (micrófono y auricular) para poder escuchar y hablar a la hora de realizar llamadas. [Consultar Apéndice II. I. Configuración del Módulo3G para la conexión a red y la gestión de llamadas.](#)
5. Comprobar el funcionamiento del Módulo 3G y del sistema montado hasta ahora. La siguiente imagen refleja cómo tiene que quedar la estructura hasta el momento, con la tableta conectada por el cable USB a Arduino y sobre éste, el Módulo 3G insertado.

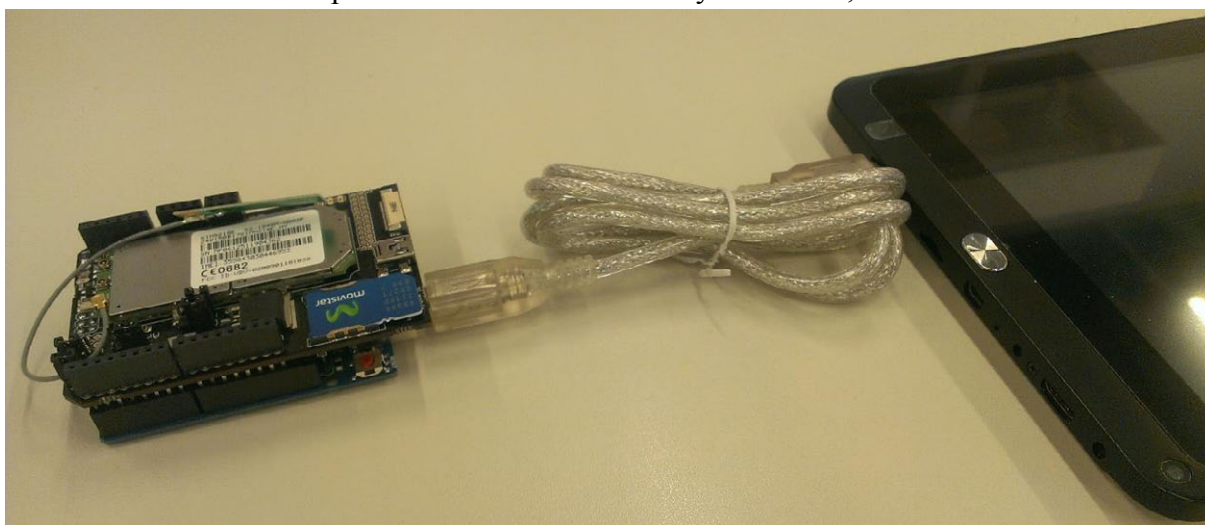


Figura 28: Tableta Android - Módulo 3G montado en Arduino

Para ello se comprueba la conexión con aplicaciones de comunicación serie desarrolladas en Android que existen en el Play Store. (21)

Dos aplicaciones que se pueden usar son: “USB Serial Monitor Lite” que es la que utiliza la propia librería “Physicaloid” ya que es el mismo creador.





Android USB Monitor Lite

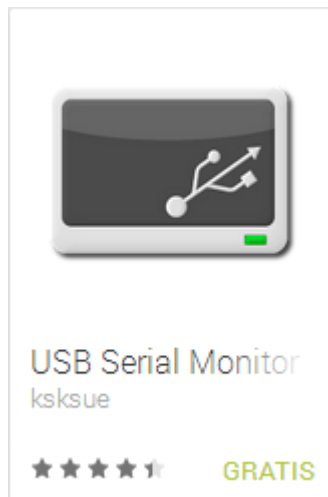


Figura 29: Icono USB Serial Monitor Lite

Para poder utilizarla se tienen que configurar las opciones de comunicación serie con los mismos valores que estén en el Módulo, la configuración queda como en la siguiente imagen.

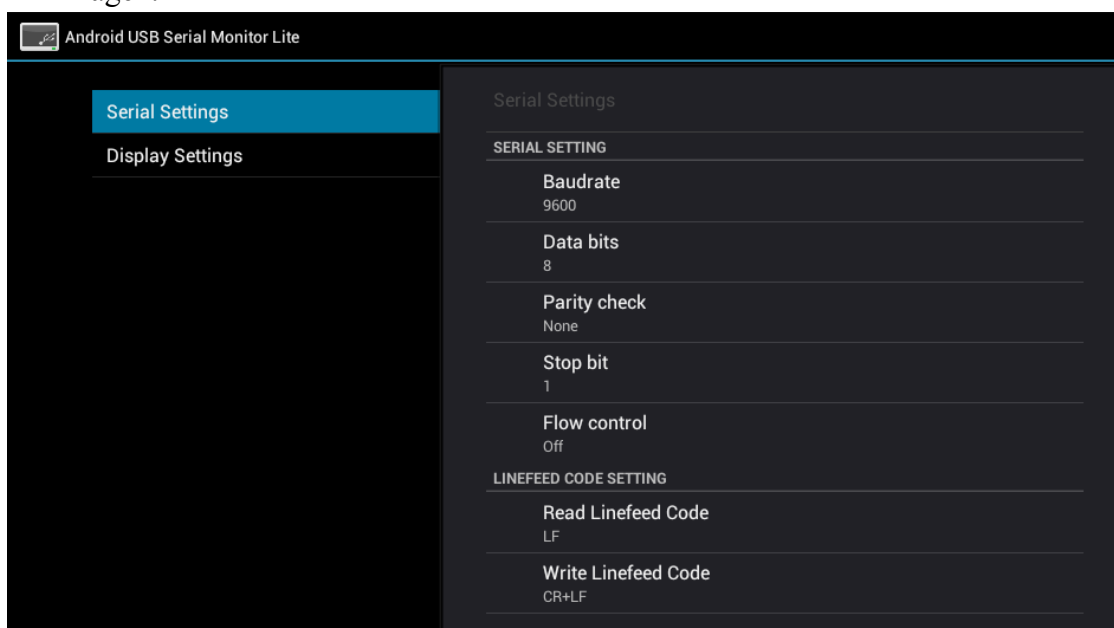


Figura 30: Captura configuración de USB Serial Monitor Lite

Una vez realizada la configuración, se efectúa una prueba como se aprecia en la siguiente figura, utilizando el simple comando “AT”, se recibe como respuesta “Ok” indicando que existe comunicación.

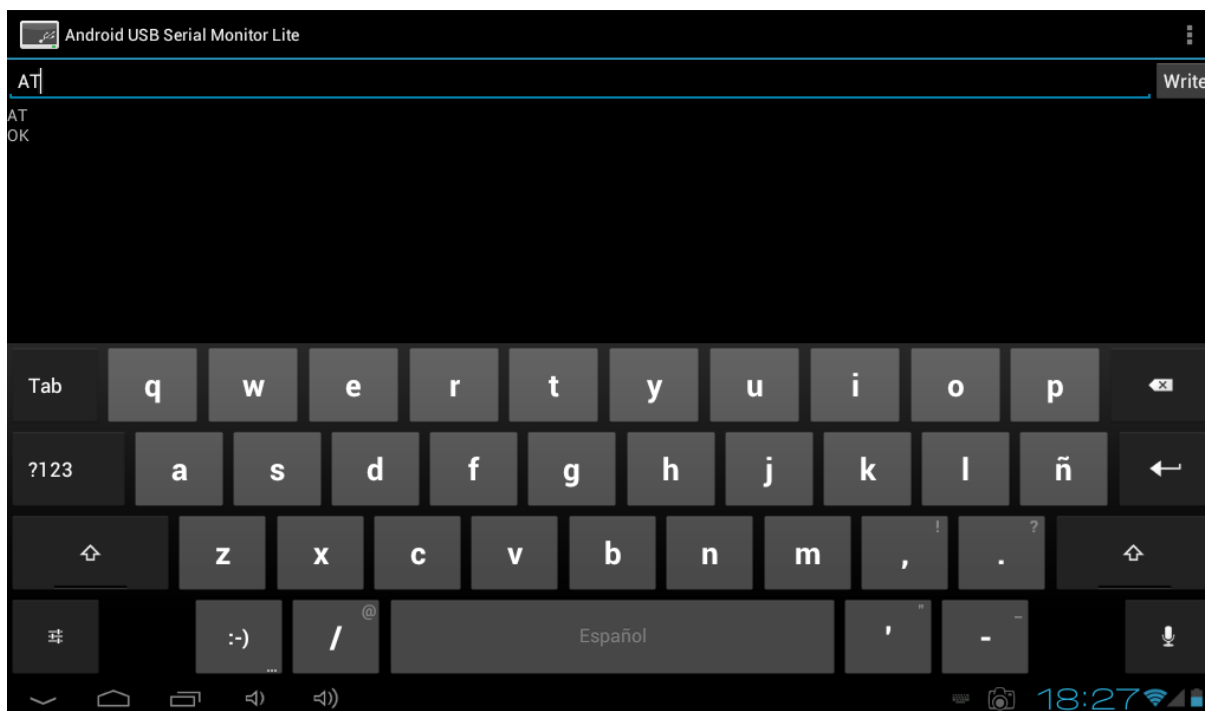


Figura 31: Captura funcionamiento de USB Serial Monitor Lite

Y la otra aplicación es “UsbTerminal” de Lior Hass.

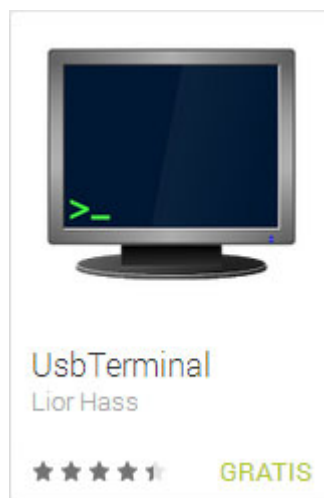


Figura 32: Icono UsbTerminal

Ésta destaca sobre la otra debido a que no tiene ningún problema en utilizar altas velocidades de transmisión (baudrates).

6. Desarrollar la aplicación Android deseada, para conseguir controlar el Módulo se usarán los comandos AT. [Consultar Apéndice II. J. Guía de comandos AT.](#)
7. Instalar y comprobar el funcionamiento.





3.2.3. Resultado final

El resultado obtenido es el de la imagen siguiente. El dispositivo Android (la tableta) conectado directamente por USB a la placa Arduino y sobre éste el Módulo 3G.

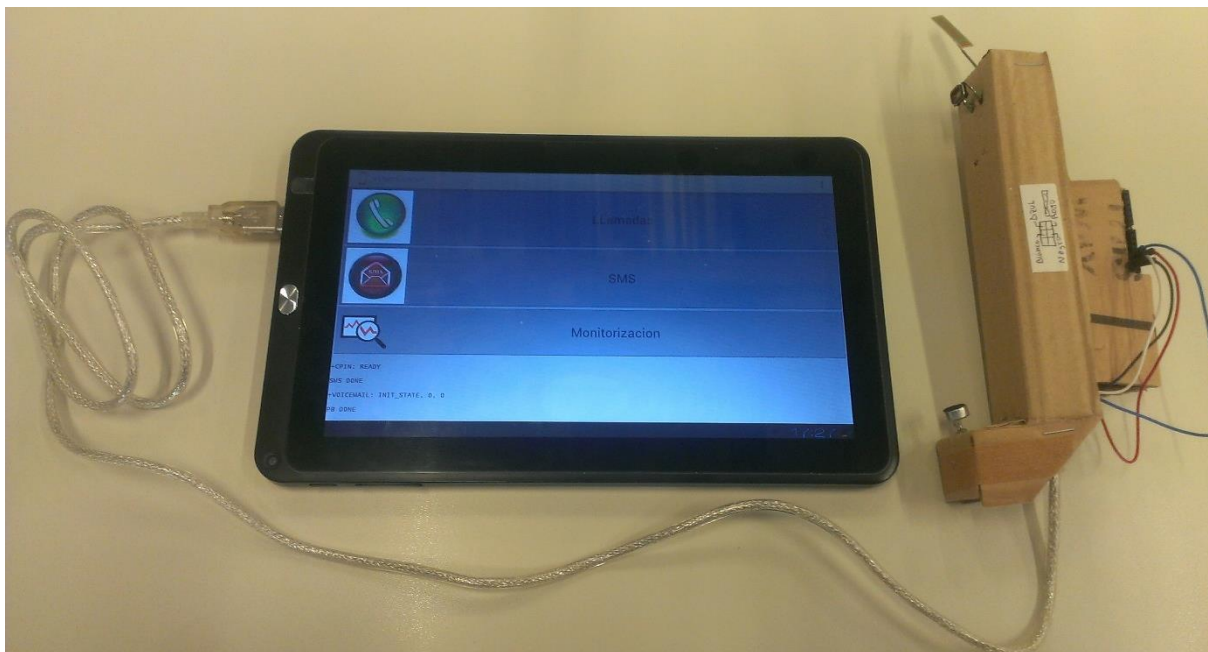


Figura 33: Resultado final del proyecto

La aplicación que maneja el Módulo 3G mediante comandos AT cuyo nombre asignado es “AppModulo3G” tiene el siguiente aspecto:

El menú principal donde se pueden escoger entre los apartados de llamadas, mensajes SMS y monitorización.



Figura 34: AppMódulo3G (Menú principal)





Para establecer conexión serie se debe pulsar la opción de “Open device” y se confirmará con un mensaje de “connected”.



Figura 35: AppMódulo3G (Menú principal conectado)

En el apartado de llamadas se pueden realizar las llamadas y recibirlas.



Figura 36: AppMódulo3G (Llamadas)



El procedimiento para realizar llamadas es introducir el teléfono de destino en el apartado correspondiente y pulsar llamar.

Para contestar una llamada, en el momento de recibir una llamada entrante aparece un mensaje de “Recibiendo llamada...” en el cual se podrá o bien descolgar, o rechazar.



Figura 37: AppMódulo3G (Recibiendo llamada)

El apartado de envío de SMS se presenta como aparece a continuación.



Figura 38: AppMódulo3G (SMS)





Para proceder al envío de un SMS se escribe el número de teléfono destino en el hueco correspondiente, se escribe el texto que se quiere enviar y finalmente se pulsa enviar.



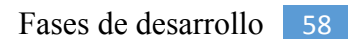
Figura 39: AppMódulo3G (Envío de SMS)

Cuando el mensaje ha sido enviado, se notifica con un aviso.



Figura 40: AppMódulo3G (SMS enviado)





AppModulo3Gv2

LISTA DE SMS RECIBIDOS
AT++CMGL="ALL"
+CMGL: 0,"REC READ","+3465027[REDACTED]",","", "14/01/23,16:16:53+04"
Hola hola hola hola hoola hola hola hola hola hola hola hola hola hola hhola hola
hola hola hola hola hola hola hola hola hola hola hola hola hola hola hola hola
+CMGL: 1,"REC READ","+3465027[REDACTED]",","", "14/01/27,18:42:23+04"
PPrueba 2 prueba 2 prueba 2
+CMGL: 2,"REC READ","+3465027[REDACTED]",","", "14/01/27,19:09:55+04"
Prueba 3
+CMGL: 3,"REC READ","+3465027[REDACTED]",","", "14/02/04,155:02:23+04"
Prueba 4
+CMGL: 4,"REC READ","77111111810511511697114",","", "14/02/08,21:49:23+04"
Infoo: Habla 24h.El 11-02 a las 20 el precio por min dde sus llamadas nacionales
dejara de ser 9,68cts IIVA inc.Recargue 5e o mas y disfrutara este preciiio 30 dias
+CMGL: 5,"REC READ","771111118105115116697114",","", "14/02/10,21:46:10+04"
Info:Habla 24h..El 11-02 a las 20 el precio/min d sus llamadas nacioonales sera

AppModulo3Gv2	
LISTA DE SMS RECIBIDOS	
AT++CMGL="ALL"	
+CMGL: 0,"REC READ","+3465027[REDACTED]", "", "14/01/23,16:16:53+04" Hola hola hola hola hoola hola hola hola hola hola hola hola hola hhola hola hola hola hola hola hola hola hola hola hola hola hola hola hola hola hola	SMS 1
+CMGL: 1,"REC READ","+3465027[REDACTED]", "", "14/01/27,18:42:23+04" PPueba 2 prueba 2 prueba 2	SMS 2
+CMGL: 2,"REC READ","+3465027[REDACTED]", "", "14/01/27,19:09:55+04" Prueba 3	SMS 3
+CMGL: 3,"REC READ","+3465027[REDACTED]", "", "14/02/04,155:02:23+04" Prueba 4	SMS 4
+CMGL: 4,"REC READ","77111111810511511697114", "", "14/02/08,21:49:23+04" Infoo: Habla 24h.El 11-02 a las 20 el precio por min dde sus llamadas nacionales dejara de ser 9,68cts IIVA inc.Recargue 5e o mas y disfrutara este precio 30 dias	SMS 5
+CMGL: 5,"REC READ","771111118105115116697114", "", "14/02/10,21:46:10+04" Info:Habla 24h..El 11-02 a las 20 el precio/min d sus llamadas nacioonales sera	SMS 6



El último apartado de “Monitorización” recoge acciones menos comunes de los teléfonos móviles.

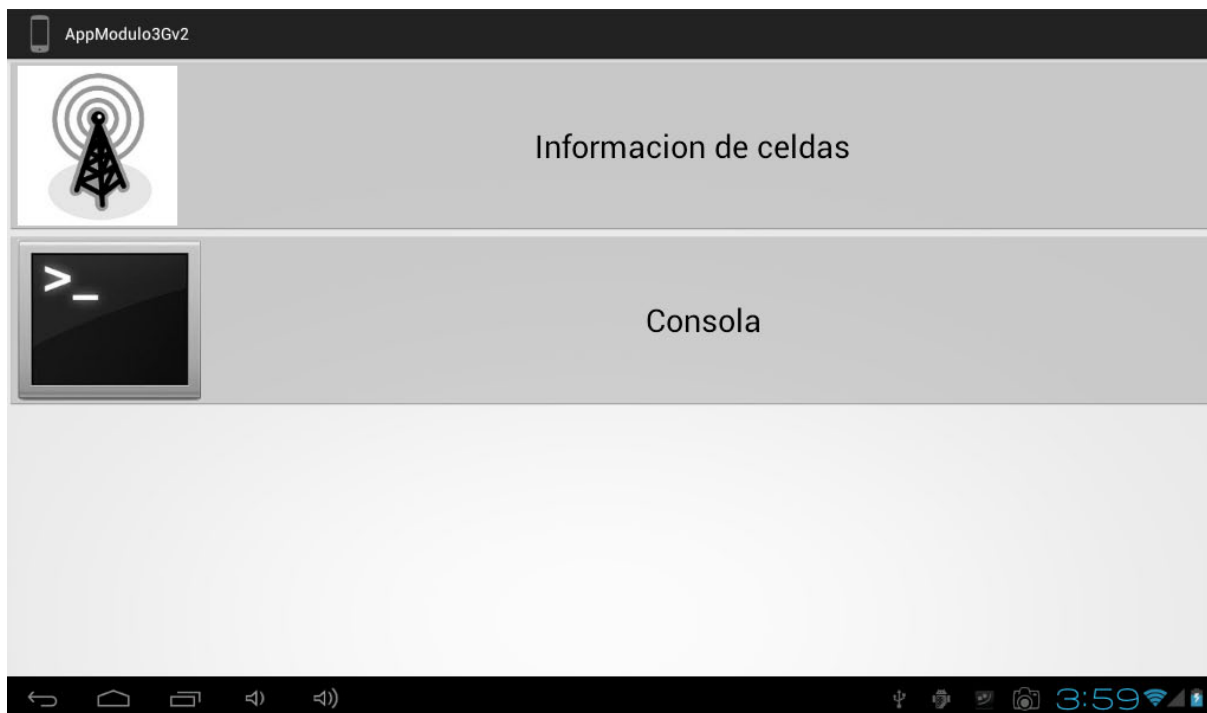


Figura 43: AppMódulo3G (Monitorización)

En “Información de celdas” se obtiene una lista con los datos de la celda servidora (SCell) y las demás celdas vecinas que están al alcance (NCell).



Figura 44: AppMódulo3G (Información de celdas)





En “Consola” se accede a un terminal donde se pueden enviar órdenes al Módulo 3G mediante comandos AT y queda registrado todo el intercambio de mensajes.

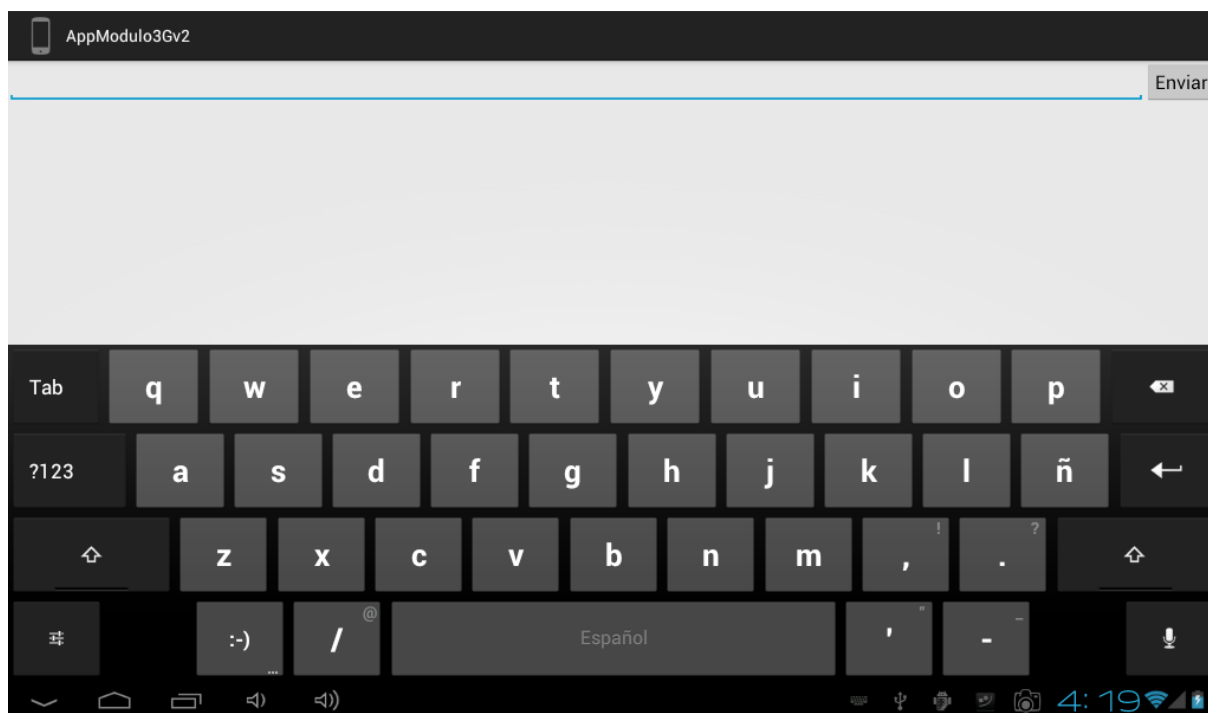


Figura 45: AppMódulo3G (Consola)

3.2.4. Inconvenientes encontrados

- El principal problema hallado, fue en relación con la librería Physicaloid, las respuestas que devolvía el Módulo en el dispositivo Android se recibían recortadas o sobrescritas, según las recibía el fallo parecía no tener un patrón y ser aleatorio. Solo ocurría en las respuestas, las solicitudes se enviaban bien.
El problema era que la velocidad de transmisión (baudrate) que viene por defecto en el Módulo de 115200bps es demasiado alta para la librería Physicaloid, ya que a altas velocidades no funciona bien.
La solución es configurar una velocidad de transmisión más baja, configurándola a 9600 bps funciona correctamente.
- Los auriculares de conexión minijack no funcionan para su uso con llamadas.
- Si se inicia la conexión entre una aplicación de la Tableta, por ejemplo android usb lite, y el Módulo 3G, para iniciar otra comunicación con otra aplicación, por ejemplo AppMódulo3G (aplicación desarrollada en el presente documento) se debe que cerrar el proceso en la Tableta de la otra aplicación, porque acapara la conexión y no se puede iniciar otra comunicación desde otra aplicación.
- Con el Módulo 3G montado en Arduino y éste enchufado a la tableta, se consume la batería de la tableta más rápido que se recarga mediante USB desde el ordenador.
- Para la obtención de información de celdas vecinas solo es posible en 2G en 3G no. No existen comandos AT que permitan la obtención de información de otras celdas en 3G.





- No hay avisos al recibir un SMS, por lo tanto, para saber si se ha recibido un nuevo mensaje se tiene que realizar una petición con el comando AT correspondiente. Para que saltara una notificación de nuevo SMS recibido, se tendría que hacer la petición periódicamente.
- Según la información que se obtiene del Módulo 3G con el comando ATI , es una Versión 2.4 pero el manual de comandos AT suministrado por Cooking hacks es de 1.21 (22), al parecer no tienen relación las versiones de placa y manual, pero de todas formas se ha encontrado una versión más reciente del manual, la versión 1.40 (23).
Pero para las acciones que se realizan cualquiera de los dos es suficiente.







4. Conclusión

Realizar este proyecto empezando casi desde cero, sin conocimientos de Arduino y programación en Android, ha sido un desafío que ha dado sus frutos. Aunque hoy en día existe una gran comunidad que comparte mucha información en la red al respecto, es necesario ponerla en contexto e integrar las diferentes tecnologías.

El resultado final del desarrollo brinda una aplicación que aunque es muy básica, ofrece una muestra de lo que es capaz de realizar el Módulo 3G, aunque no se explotan todas las capacidades de la placa. El objetivo principal consistente en desarrollar una aplicación de Android con la cual se pudiese controlar el Módulo 3G vía USB se ha cumplido.

Todo el procedimiento queda plasmado de forma que se pueda consultar como un manual a la hora de iniciarse con los conceptos tratados.

Ha sido una experiencia gratificante y enriquecedora ya que durante el transcurso del proyecto se adquieren y se integran conocimientos de muy diferentes campos que son bastante útiles hoy en día en el mundo laboral debido al auge que existe de las nuevas tecnologías y proporcionan una visión de conjunto de la que se carece durante la etapa de estudio de las diferentes asignaturas de la carrera.

5. Líneas futuras de desarrollo

En principio, este proyecto está concebido como una introducción al manejo y control del Módulo 3G desde Android, destinado a todo aquél que quiera continuar e ir mejorando la aplicación o desarrollar la suya propia. También sirve como base para aquellos que solo vayan a usar Arduino o le vayan a colocar otro módulo.

Sólo se ha mostrado un ejemplo de las funcionalidades del Módulo 3G, gracias a que se puede controlar mediante la larga lista de comandos AT, es posible desarrollar infinidad de aplicaciones a gusto del consumidor.

Aún quedan facultades que no se han mostrado como el GPS, el uso del Módulo 3G como módem, el uso de la cámara con video llamadas, reproducción de música, correos electrónicos, FTP y FTPS, TCP y UDP, HTTP y HTTPS.

Mediante comandos AT es posible situar el módulo 3G en modo depuración. En este modo proporciona mucha información acerca de los procedimientos y métodos de funcionamiento de la interfaz radio. Aunque el formato de los comandos y de la información proporcionada no es de dominio público hay trabajos que intentan mediante ingeniería inversa decodificar esta interfaz y proporcionar un acceso público a la misma.

Por lo tanto queda abierto un gran abanico de posibilidades para desarrollar.







6. Apéndice I (Relativo a Arduino)

A. Como saber si el dispositivo Android tiene soporte USB Host Mode³

Algunos dispositivos Android pueden actuar como un host USB, de modo que se pueden utilizar otros dispositivos USB conectados a ellos. Por ejemplo, explorar o importar las fotografías almacenadas en una cámara o copiar archivos en un dispositivo de memoria USB conectado a la tableta, conectando un teclado, mouse USB, GPS externo o un dispositivo Wi-Fi.

Cuando el dispositivo Android actúa como un host USB, se encarga de alimentar el bus. Cuando el dispositivo Android está en el modo Accesory, el bus se alimenta por el equipo conectado. El USB Host Mode y USB Accesory son compatibles con Android empezando desde la versión 3.1 y superiores, soportando una amplia gama de dispositivos, por otro lado, los fabricantes de hardware deciden por su cuenta si quieren añadir estas bibliotecas a sus productos. Para utilizar un dispositivo USB conectado se tiene que disponer de:

- Un cable adaptador USB OTG (USB On-The-Go) en caso de que el dispositivo Android no tenga un puerto USB tipo A. Normalmente traen un puerto micro USB, en estos casos sí es necesario el adaptador. [Más información en Apéndice I: B. Cable USB.](#)
- Drivers USB Host Mode cargados en el dispositivo.
- Android a partir de la versión 3.1 inclusive, para que soporte el modo USB Host.

Hay algunas aplicaciones que pueden ayudar a comprobar que se tiene todo el hardware y software necesario. La aplicación que se utiliza es “USB Host Diagnostics” disponible en Google Play (21). Para probar las capacidades del dispositivo se han de seguir los siguientes pasos:

- Instalar USB Host Diagnostics desde Play Store.
- Asegurar que no hay nada conectado al puerto USB.
- Ejecutar USB Host Diagnostic y seguir las instrucciones.
- Conectar un dispositivo (Arduino, unidad flash USB, teclado, ratón, etc.) cuando lo solicite.

Finalmente, aparece una ventana con un resumen de sobre el hardware detectado, el modo de ejecución, etc, permitiendo conocer el nivel de compatibilidad y estado del USB Host Mode.

³ Consultar Apéndice I: C. Modos USB Host y Accesory de Android.





B. Cable USB



Figura 46: Logo USB OTG

Para poder conectar al dispositivo Android otros dispositivos USB (ratones, memorias, teclados, Arduinos...) es necesario utilizar el modo USB host y para ello la tableta tiene que disponer de un puerto USB tipo A que alimente con 5v el bus, en caso de que no lo tenga, en vez de éste, tendrá un puerto micro USB o mini USB, pero para que éstos puedan alimentar el bus, es imprescindible un cable adaptador OTG ⁴ (On The Go) (24).



Figura 47: Adaptador OTG (micro USB-USB tipo A)

Normalmente los dispositivos con Android como los móviles o las tabletas llevan implantados un conector micro USB, en tal caso es necesario el cable OTG, al cual enchufaremos el cable micro USB que se posee.

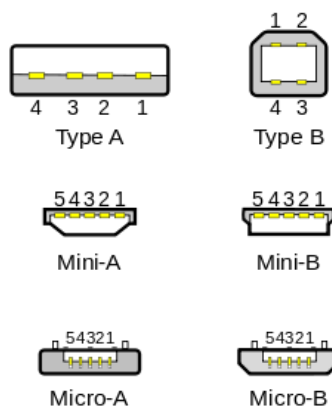


Figura 48: Tipos de USB

⁴ USB On-The-Go, también conocido por el acrónimo USB OTG y como USB Host, es una extensión de la norma USB 2.0 que permite a los dispositivos USB tener más flexibilidad en la gestión de la conexión USB. Permite que dispositivos como un reproductor de audio digital o teléfono móvil actúen como host, por lo que se les puede conectar una memoria USB, un ratón, un teclado, un disco duro, un módem, etc.



En la siguiente figura se puede apreciar como es necesario un cable USB OTG para conectar un teléfono móvil a una plataforma Arduino.



Figura 49: Android conectado con OTG a Arduino (25)

En éste proyecto no es necesario el cable USB OTG ya que la tableta dispone de un puerto USB tipo A.

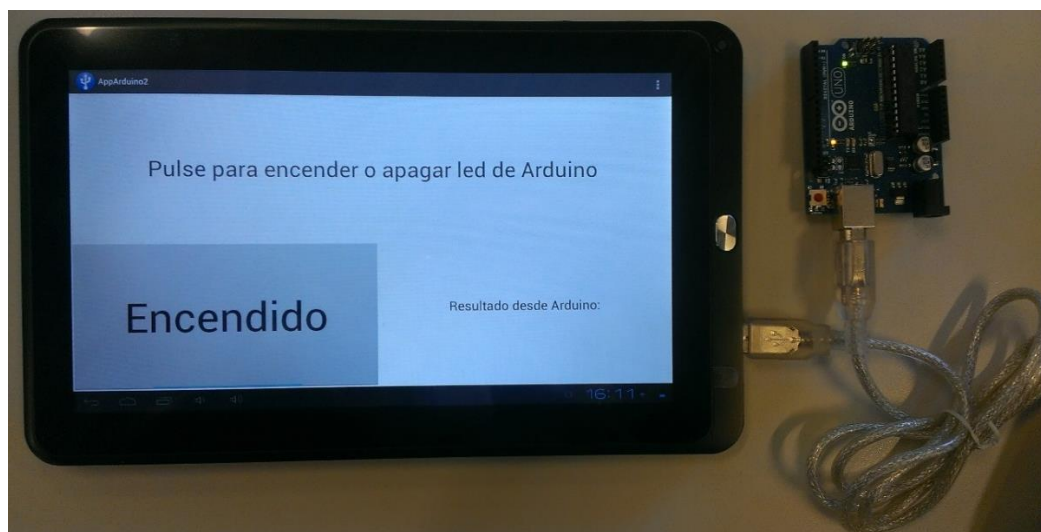


Figura 50: Imagen de la conexión Arduino – tableta vía USB

C. Modos USB Host y Accessory de Android.

Android es compatible con una gran variedad de periféricos USB y accesorios USB a través de dos modalidades: USB Accessory y USB Host. (26)

En el modo USB Accessory, el hardware externo USB actúa como host. Ejemplos de accesorios podrían incluir controladores de robótica, estaciones de conexión, equipos de diagnóstico y musical, quioscos, lectores de tarjetas, y mucho más. Esto le da a los dispositivos con Android que no cuentan con las capacidades de acogida la capacidad de interactuar con el hardware USB. Los accesorios USB de Android deben estar diseñados para trabajar con dispositivos con Android y deben cumplir con el protocolo de comunicación de accesorios Android (Android accessory communication protocol).

En el modo USB Host, el dispositivo con Android actúa como host. Ejemplos de dispositivos incluyen cámaras digitales, teclados, ratones y dispositivos de juego. Dispositivos USB que están diseñados para una amplia gama de aplicaciones y entornos aún pueden interactuar con las aplicaciones Android que se pueden comunicar correctamente con el dispositivo.

La figura que se presenta a continuación muestra las diferencias entre los dos modos. Cuando el dispositivo con Android está en modo USB Host, actúa como host y alimenta al bus. Cuando el dispositivo con Android está en el modo USB Accessory, el hardware USB conectado funciona como el host y alimenta al bus.

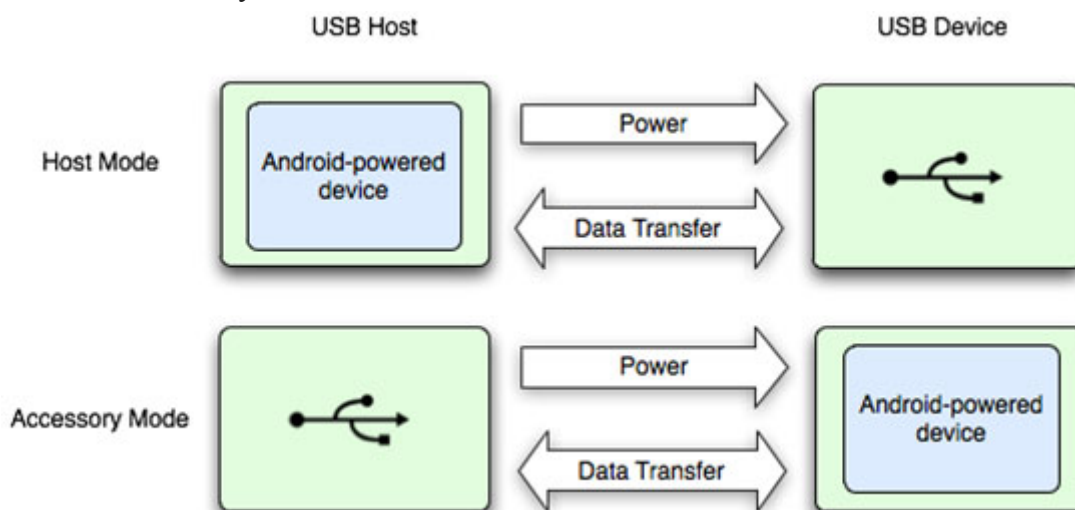


Figura 51: Modos USB Host y Accessory

Los modos USB Accessory y USB host son compatibles desde Android 3.1 (nivel de API 12). El modo USB Accessory también será portado a Android 2.3.4 (nivel de API 10) como una biblioteca de complemento para apoyar una amplia gama de dispositivos. Los fabricantes de hardware decidirán por su cuenta si quieren añadir estas bibliotecas a la imagen del sistema.



D. Guía de instalación de ADB drivers en PC

Android Debug Bridge (ADB) es una de las herramientas incluidas en el Android SDK que facilita la comunicación entre el dispositivo Android y el ordenador. Además ofrece una interfaz de línea de comandos donde se pueden ejecutar una serie de comandos para realizar determinadas tareas en el dispositivo.

A. Cómo instalar los ADB driver en Windows 7, 8, Vista & XP:

Antes de empezar:

- Es necesario tener Java instalado en el ordenador.
- Asegurarse de tener el modo USB debugging encendido en el dispositivo Android.

- 1) Descargar e instalar Android SDK, se creará un directorio llamado “android-sdk-windows”
- 2) Abrir el directorio “android-sdk-windows” y ejecutar “SDK Manager.exe”
- 3) Seleccionar los siguientes paquetes: Android SDK Platform –tools y Google USB Driver, botón derecho e instalar.
- 4) En la siguiente ventana seleccionar la opción Accept All y seleccionar el botón de Install.
- 5) Conectar el dispositivo al PC, todavía no se reconocerá.
- 6) Abrir el Administrador de dispositivos y localizarlo bajo “Otros dispositivos”.
- 7) Botón derecho sobre él y clicar “Actualizar software de controlador”.
- 8) Pulsar en “buscar software de controlador en el equipo”.
- 9) En la siguiente pantalla escoger “Elegir una lista de controladores de dispositivo en el equipo”.
- 10) Seleccionar “Mostrar todos los dispositivos” y seleccionar siguiente.
- 11) Pulsar “Usar disco...” y en la ventana que aparecerá, explorar hasta seleccionar el fichero: android-sdk-windows\extras\google\usb_driver\android_winusb.inf
- 12) Seleccionar la opción “Android Composite ADB Interface” y pulsar siguiente.
- 13) Confirmar la instalación clicando “Sí” e “Instalar”.
- 14) Android ADB drivers están ahora instalados en Windows para el dispositivo. Aparecerá un mensaje de confirmación.

B. Cómo comprobar que se han instalado correctamente

- 1) Conectar el dispositivo al PC.
- 2) Desplazarse hasta el directorio: android-sdk-windows\platform-tools.
- 3) Mantener Shift y presionar botón derecho.
- 4) Seleccionar “Abrir ventana de comandos aquí”.
- 5) Ejecutar el comando: adb devices.
- 6) Si la ID del dispositivo aparece en la pantalla, se han instalado correctamente.





E. Guía de instalación de Arduino en Windows

1) *Obtener un Arduino y un cable USB*

Este tutorial está orientado para la placa Arduino UNO, Duemilanove, Nano, Arduino Mega 2560 o Diecimila.



Figura 52: Arduino UNO

También se necesita un cable estándar USB (conexión A a conexión B). En el caso de la placa Arduino Nano se necesita un cable de conexión A a conexión mini-B.



Figura 53: Cable estándar USB

2) *Descargar el IDE de Arduino*

Descargar la última versión de la página de descargas oficial de Arduino, descomprimir el fichero asegurándose de mantener la estructura de directorios. Abriendo la carpeta arduino-00XX se deberían ver dentro una serie de ficheros y carpetas.

3) *Conectar la placa*

Conectar la placa Arduino al ordenador usando el cable USB, el LED verde indicador de la alimentación (nombrado como PWR en la placa) debería quedar encendido a partir de ese momento.

Si es una placa Arduino Diecimila, asegurarse de que la placa está configurada para alimentarse mediante la conexión USB. La fuente de alimentación se selecciona con un puente ("jumper"), una pequeña pieza de plástico que encaja en dos de los tres pines situados entre los



conectores USB y de alimentación de la placa. Comprobar que el puente esté conectando los dos pines más cercanos al puerto USB de la placa.

En las placas Arduino UNO, Arduino Duemilanove y Arduino Nano la fuente de alimentación adecuada se selecciona de forma automática y no requiere de realizar ninguna comprobación en este sentido.

4) *Instalar los drivers*

Instalación de controladores para Arduino Uno o Arduino Mega 2560 con Windows 7, Vista o XP:

Conectar la placa y esperar a que Windows inicie el proceso de instalación del controlador. Después de unos momentos, el proceso fallará.

Hacer clic en el menú Inicio y abrir el Panel de control.

En el Panel de control, ir a Sistema y seguridad. A continuación, clic en Sistema, abrir el Administrador de dispositivos.

Buscar debajo de Puertos (COM y LPT). Se debe ver un puerto abierto llamado "Arduino UNO (COMxx)"

Hacer clic derecho en el "UNO Arduino (COMxx) " puerto y elegir la opción "Actualizar software de controlador".

A continuación, seleccionar la opción "Buscar software de controlador en el equipo".

Por último, buscar y seleccionar el archivo controlador llamado "arduino.inf", ubicado en la carpeta "Drivers" de la descarga del software de Arduino (no el "de los controladores USB FTDI" sub - directorio). Si se está utilizando una versión antigua del IDE (1.0.3 o anterior), seleccionar el archivo controlador del Uno llamado "Arduino UNO.inf".

Windows terminará la instalación del controlador desde aquí en adelante.

Se puede comprobar que los *drivers* se han instalado correctamente abriendo la carpeta del Administrador del Dispositivos, en el grupo *Dispositivos* el panel de control del sistema. Buscar "USB Serial Port" (o *Puerto USB-Serie*) en la sección de puertos; esa es la placa Arduino.

5) *Ejecutar la Aplicación Arduino*

Hacer doble clic en la aplicación Arduino.

6) *Abrir el ejemplo Blink*

Abrir el programa de ejemplo para hacer parpadear un LED ("LED blink"): Archivo > Ejemplos > 1.Basics > Blink.



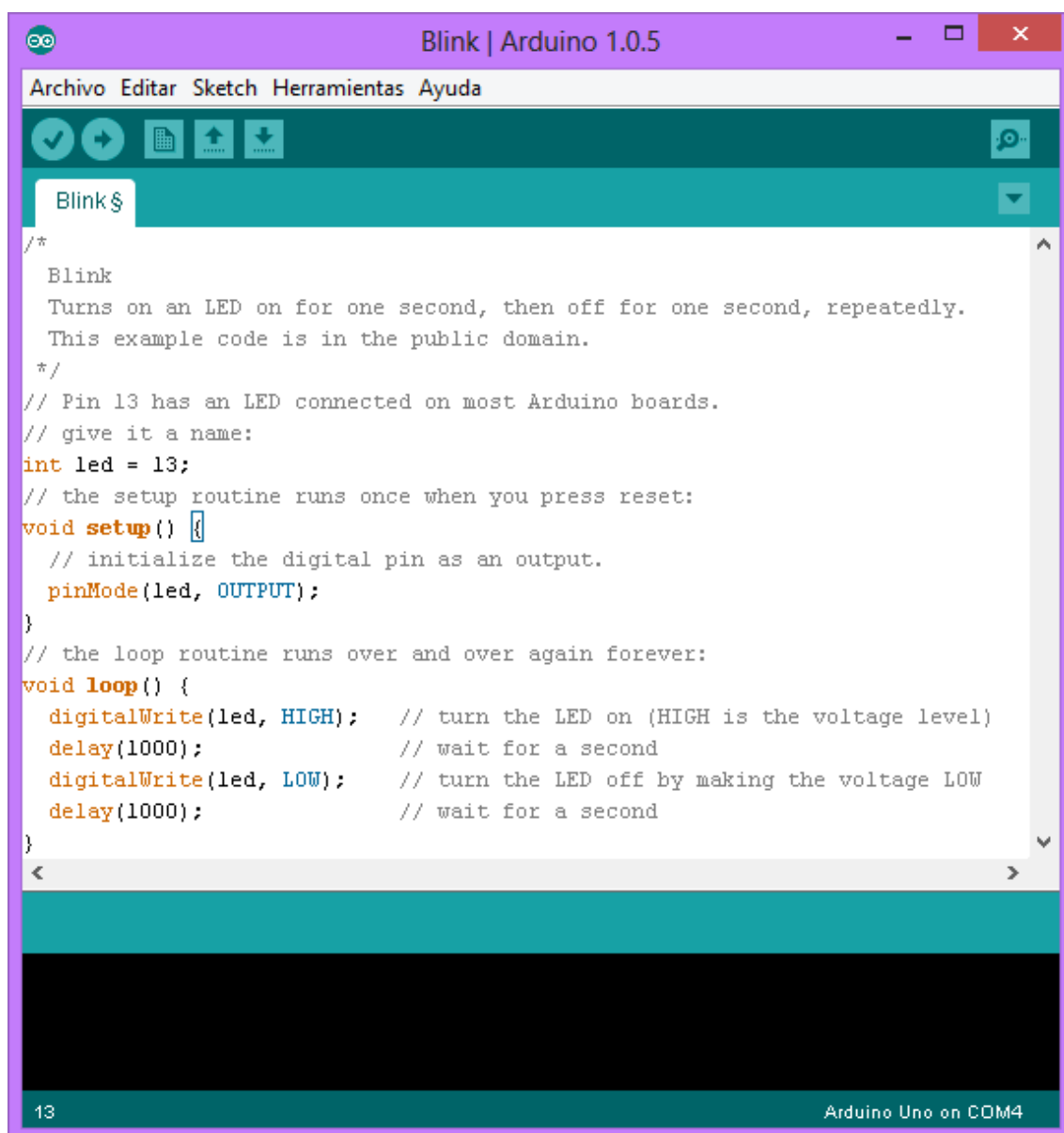


Figura 54: Ejemplo de Sketch Blink

7) Seleccionar la placa

Seleccionar el tipo de placa del Arduino en el menú Herramientas > Tarjeta.

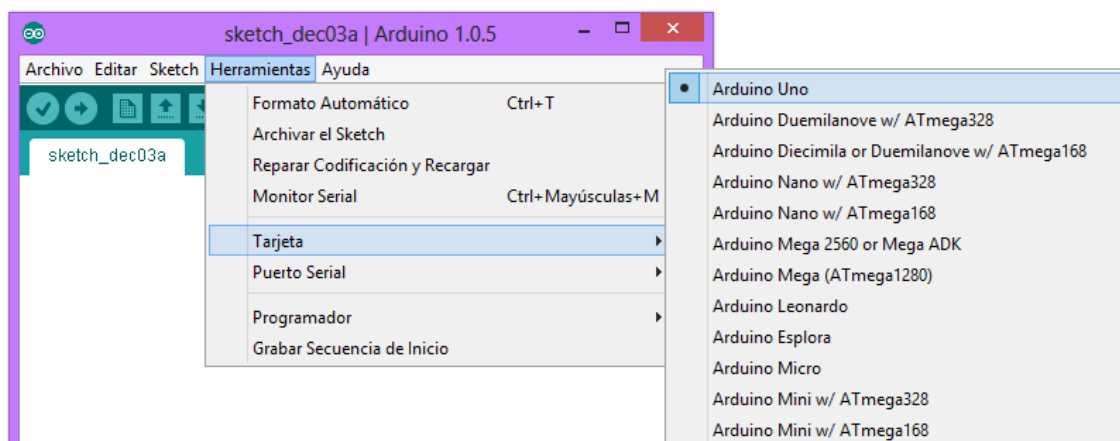


Figura 55: seleccionar placa Arduino





8) *Seleccionar el puerto serie*

Seleccionar el dispositivo serie de la placa Arduino en el menú Tools | Serial Port (*Herramientas | Puertos Serie*). Lo más probable es que sea COM3 o mayor (COM1 y COM2 se reservan, por regla general para puertos serie de hardware). Para asegurarse de cuál es, se puede desconectar la placa y volver a mirar el menú; el puerto de la placa habrá desaparecido de la lista. Reconectar la placa y seleccionar el puerto apropiado.

9) *Subir el sketch a la placa*

Pulsar sobre el botón "Upload" en el Entorno Arduino. Esperar unos pocos segundos, se debería ver parpadear los led RX y TX de la placa. Si el volcado del código es exitoso se verá aparecer el mensaje "Done uploading" en la barra de estado. (*Aviso: Si se tiene una placa Arduino Mini, NG, u otras placas, se necesitará presionar el botón de reseteo de la placa inmediatamente antes de presionar el botón "Upload" el Entorno de programación Arduino.*)

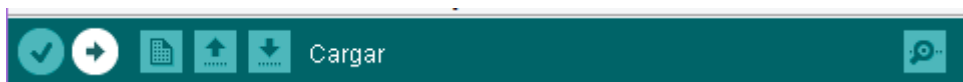


Figura 56: Opción cargar

Unos pocos segundos después de finalizar el volcado del programa se debería ver cómo el led de la placa conectado al pin 13 (L) comienza a parpadear (con un color naranja). Si ocurre esto ya se tiene Arduino listo y funcionando.





F. Guía de instalación de Android en Windows

Configuración del paquete ADT

El paquete ADT proporciona todo lo necesario para empezar a desarrollar aplicaciones, incluyendo los componentes esenciales del SDK de Android y una versión del Eclipse IDE (Integrated Development Integrated) incluyendo el ADT para agilizar el desarrollo de la aplicación Android (27).

Instalación del SDK y Eclipse IDE

1. Descomprimir el fichero ZIP (llamado `adt-bundle-<os_platform>.zip`) y guardarlo en un lugar apropiado, como por ejemplo en un directorio “Desarrollo” en el directorio personal.
2. Abrir el directorio `adt-bundle-<os_platform>/eclipse/` y ejecutar Eclipse.

Ya está, el IDE está cargado con el complemento ADT y el SDK está listo para funcionar.

Para empezar el desarrollo, se puede aprender mediante los tutoriales que ofrece la propia web de desarrolladores oficial de Android (28).

Información adicional

A medida que se continúe el desarrollo de aplicaciones, es posible que se tenga que instalar versiones adicionales de Android para el emulador y otros paquetes. Para instalar más paquetes, se utiliza el SDK Manager.

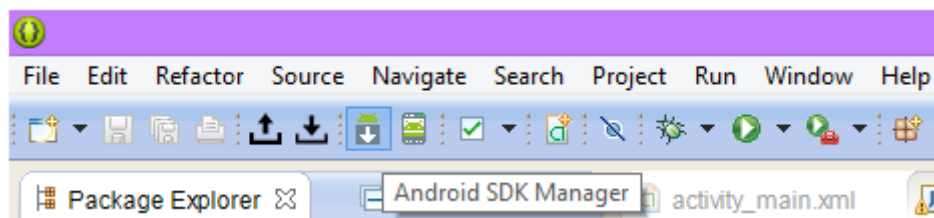


Figura 57: Ubicación del SDK Manager en Eclipse

Todo lo necesario para desarrollar aplicaciones Android se encuentra en la web oficial de desarrolladores de Android (7).



7. Apéndice II (Relativo al Módulo 3G)

G. Instalación del Módulo 3G para Windows 8

En esta ocasión se procede a instalar el software necesario para que el Sistema Operativo Windows 8 reconozca el dispositivo, pero para otros sistemas operativos es posible que no haga falta ya que se reconocerá automáticamente, aun así en la web de la tienda se proporciona indicaciones para la instalación en otros Sistemas operativos (29).

A continuación se explican los pasos a seguir para instalar el software que requiere el SIM5218.

Enchufar el Módulo 3G directamente al ordenador mediante el cable USB.



Figura 58: Módulo 3G enchufado directamente al PC

Aparecerá un mensaje de Windows “instalando software automáticamente” y no lo conseguirá, es normal.

Descargar los drivers necesarios para el correcto funcionamiento del dispositivo (30) y descomprimir en un directorio de libre elección.

Abrir el Administrador de dispositivos, en el apartado “Otros dispositivos” aparecerán los dispositivos que no se reconocen ya que necesitan un driver y por lo tanto el sistema operativo no los puede detectar. Tendrán que aparecer cinco dispositivos con el nombre “SimTech SIM5218”, que son los que pertenecen al Módulo 3G.

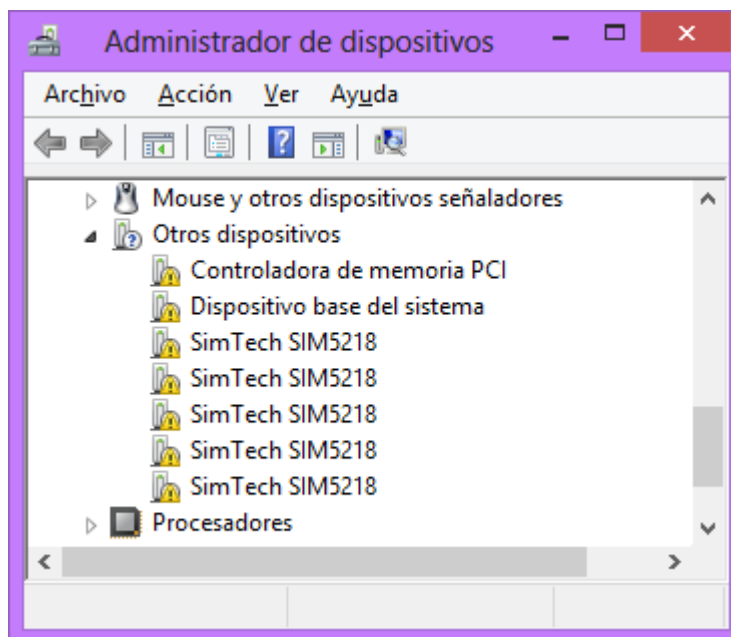


Figura 59: Dispositivos sin controlador en panel de control

Antes de instalar los drivers para el módulo SIM5218, comprobar que versión de Sistema Operativo se tiene. Dependiendo de la versión se utilizarán los drivers de 32 o 64 bits.

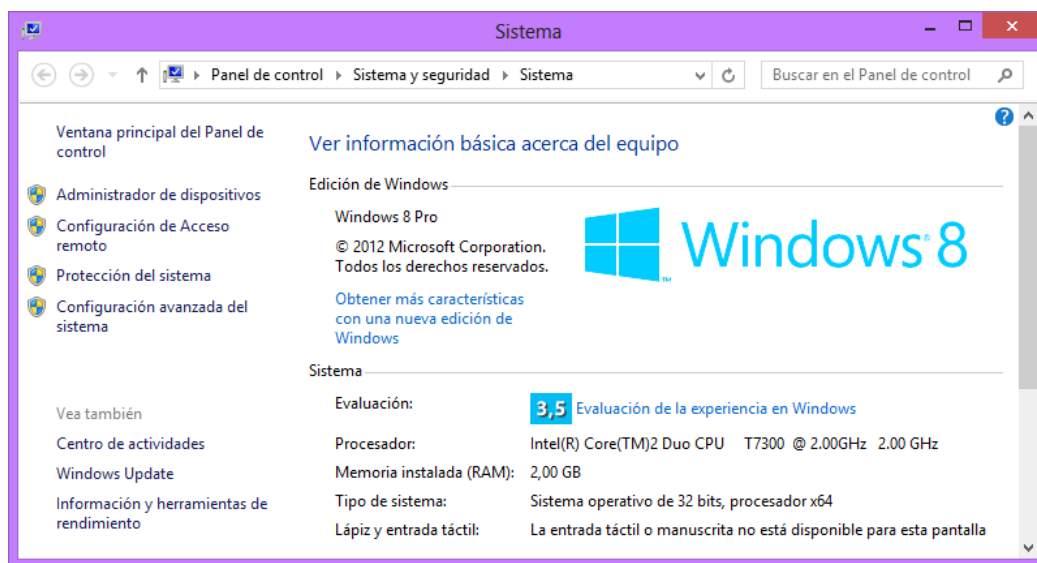


Figura 60: Propiedades de Windows





Para instalar los drivers, en el Administrador de dispositivos, en cada uno de los cinco dispositivos SimTech SIM5218 que aparecen repetir la acción de: botón derecho -> Actualizar software de controlador...

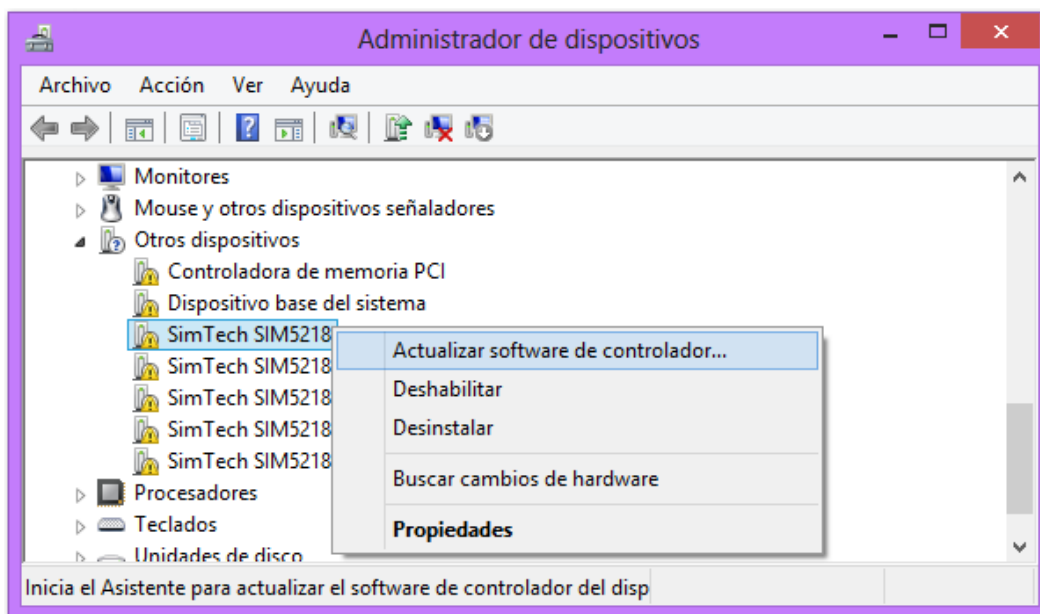


Figura 61: Actualizar controlador

Buscar la ubicación de los controladores descargados.

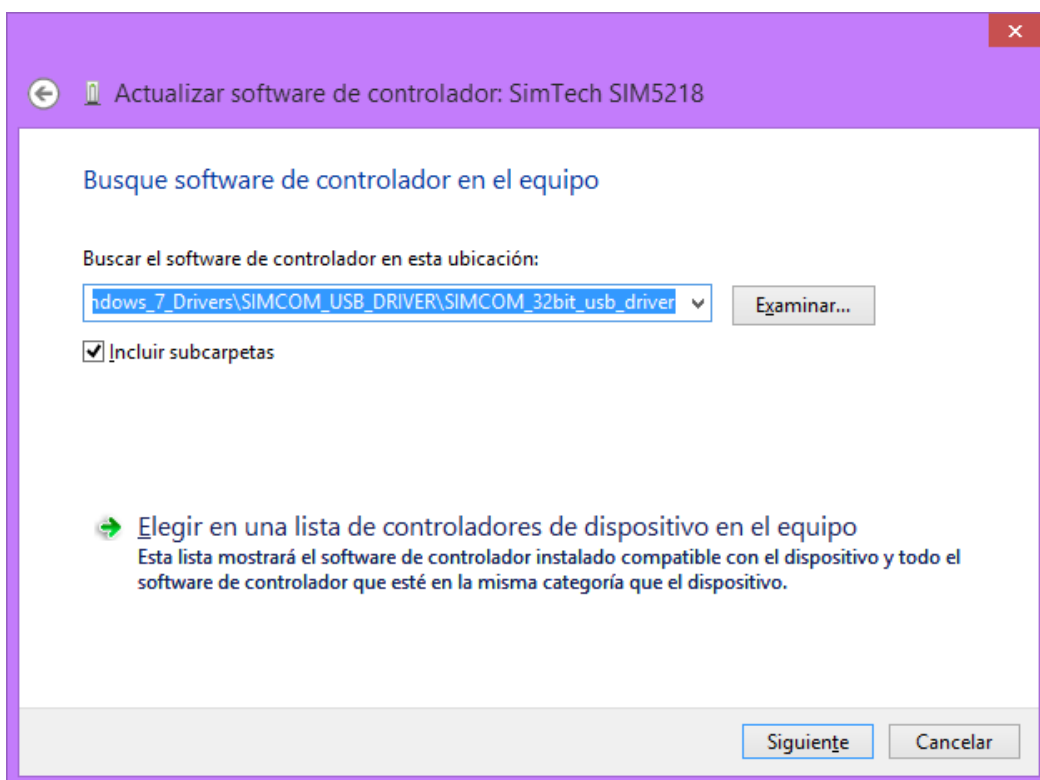


Figura 62: Buscar controlador



Aceptar y los drivers se instalarán. Aparecerá en cada uno de los dispositivos las siguientes ventanas al finalizar la instalación.



Figura 63: Ventanas de drivers instalados del Módulo 3G



Y en el Administrador de Dispositivos aparecerán como a continuación, dos en Módems y tres en Puertos COM.

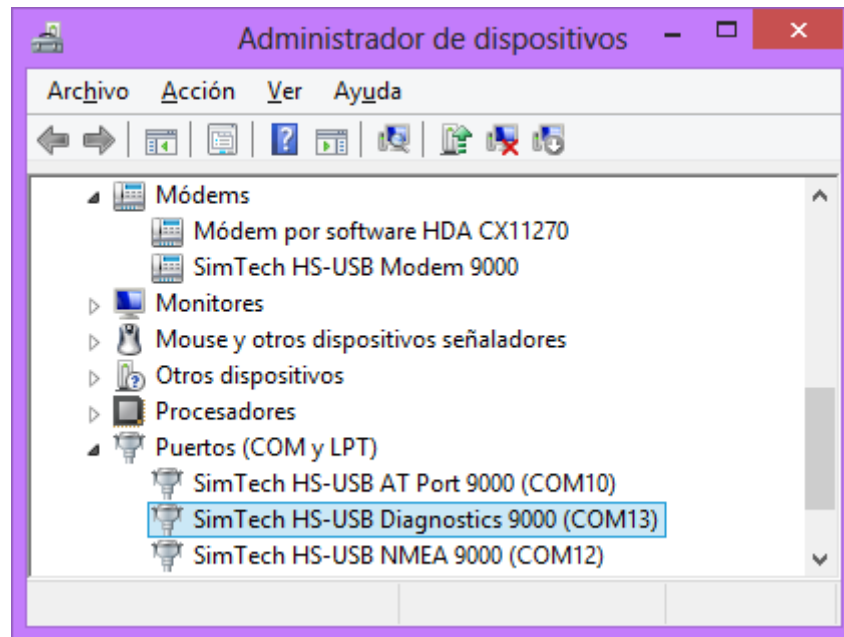


Figura 64: Dispositivos del Módulo 3G instalados

Se debe recordar que el dispositivo que aparece arriba como “SimTech HS-USB AT Port 9000” con el puerto COM10, es el encargado de la comunicación mediante comandos AT, y se tendrá que seleccionar para tal fin.





H. Introducción a HyperTerminal

Para comunicarse con el módulo SIM5218 desde el ordenador mediante comandos AT es necesario el uso de un programa que permita una conexión serial a través del puerto USB como por ejemplo HyperTerminal (Windows) o cutecom/gtkterm (Linux), el monitor serial del entorno Arduino también es válido.

HyperTerminal se incorpora de serie con las versiones de Windows XP e inferiores, en Windows 7 y 8 no se incluye, pero puede descargarse e instalarse (31).

HyperTerminal es un programa que se puede utilizar para conectar con otros equipos, sitios Telnet, sistemas de boletines electrónicos (BBS), servicios en línea y equipos host, mediante un módem, un cable de módem nulo o Ethernet.

Sirve para configurar y probar el módem o examinar la conexión con otros sitios, graba los mensajes enviados o recibidos por servicios o equipos situados al otro extremo de la conexión, permite revisar el texto recibido que sobrepase el espacio de la pantalla.

Una vez instalado HyperTerminal, crear una nueva conexión. Se le asigna un nombre, en este caso “AT_Command”.

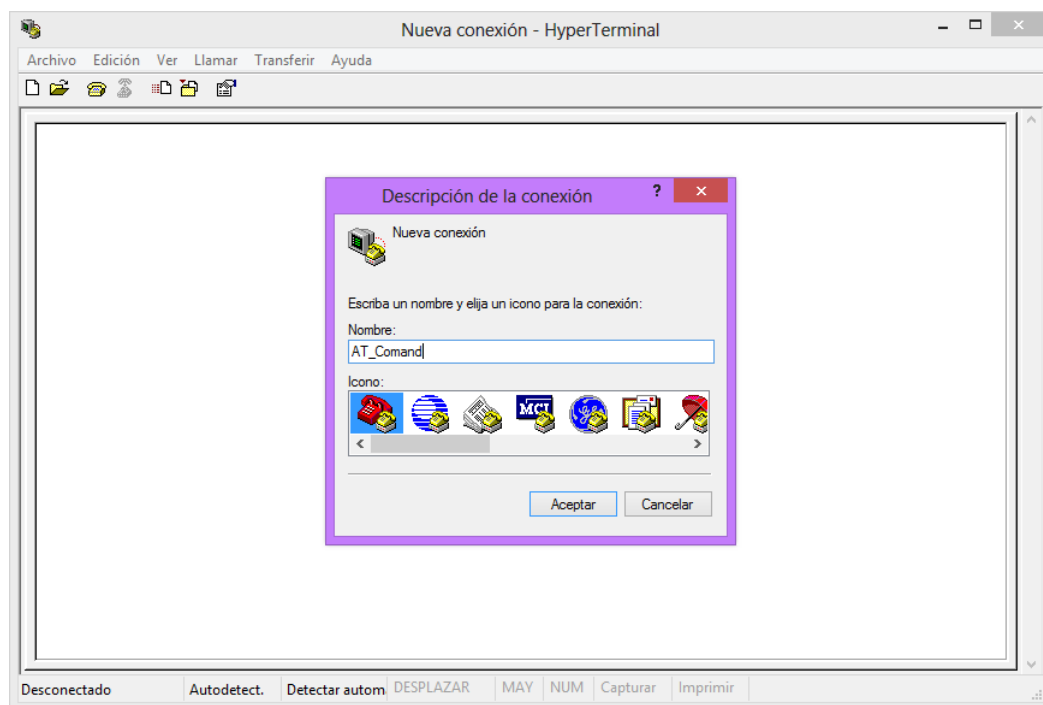


Figura 65: Hyper Terminal

El puerto a escoger para comunicarse con el Módulo 3G mediante comandos AT es el que aparezca en el Administrador de dispositivos de Windows con el nombre “SimTech HS-USB AT Port 9000”, que en éste caso es el puerto COM10.

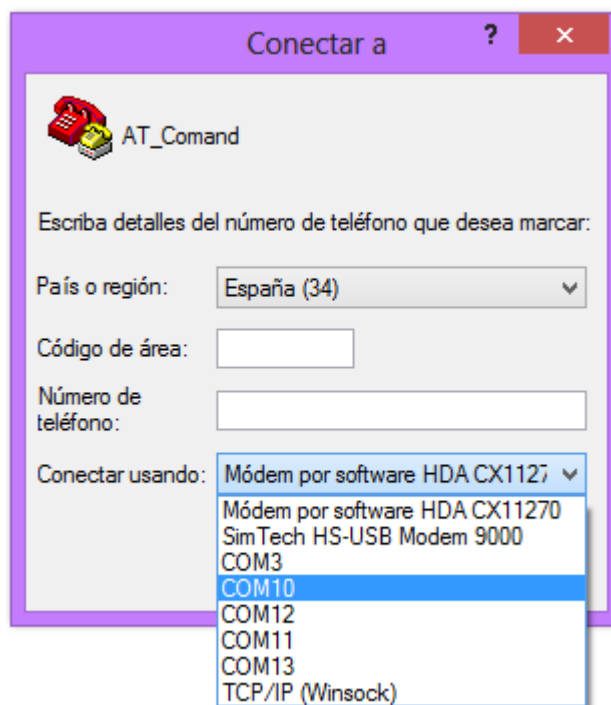


Figura 66: Nueva conexión (HyperTerminal)

Para lograr la comunicación se debe configurar la conexión como en la siguiente figura.

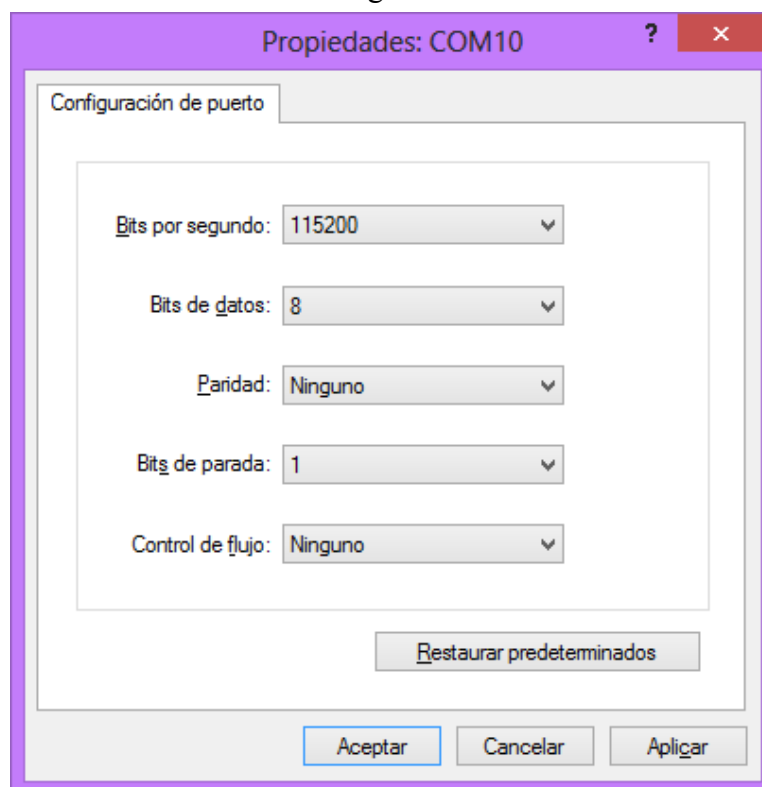


Figura 67: Configuración (HyperTerminal)

Por defecto el módulo Sim5218 viene configurado a 115200 bps. Para probar la funcionalidad del módulo desde el ordenador lo dejaremos así, pero para que funcione con Android se debe bajar la velocidad de transmisión a 9600 bps, cuestión explicada en el apartado de la librería Physicaloid.





Después de haber configurado las opciones, abrir el puerto serie, y encender el Módulo 3G presionando el botón de encendido durante unos dos segundos.



Figura 68: Detalle del botón de encendido/apagado del Módulo

Si en el momento de probarlo no hay tarjeta SIM introducida, aparecerá este mensaje lo primero.

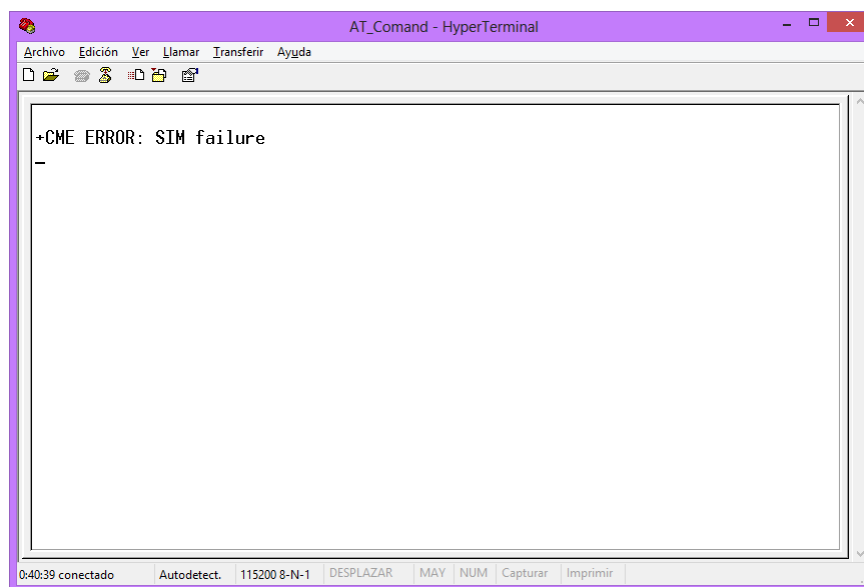


Figura 69: Error por ausencia de SIM (HyperTerminal)





Para comprobar que realmente existe comunicación se utiliza el comando AT básico: “AT” y se tiene que obtener la respuesta “OK”. Es indiferente que se escriba en mayúsculas o en minúsculas.

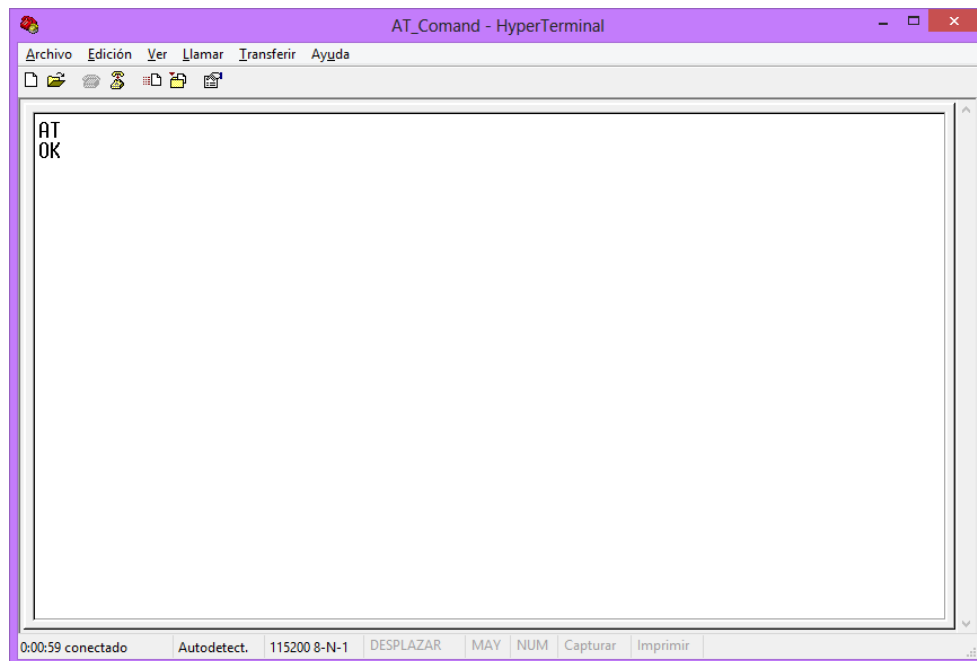


Figura 70: Conexión correcta (HyperTerminal)

Probando otro comando AT que devuelve la información del Módulo 3G

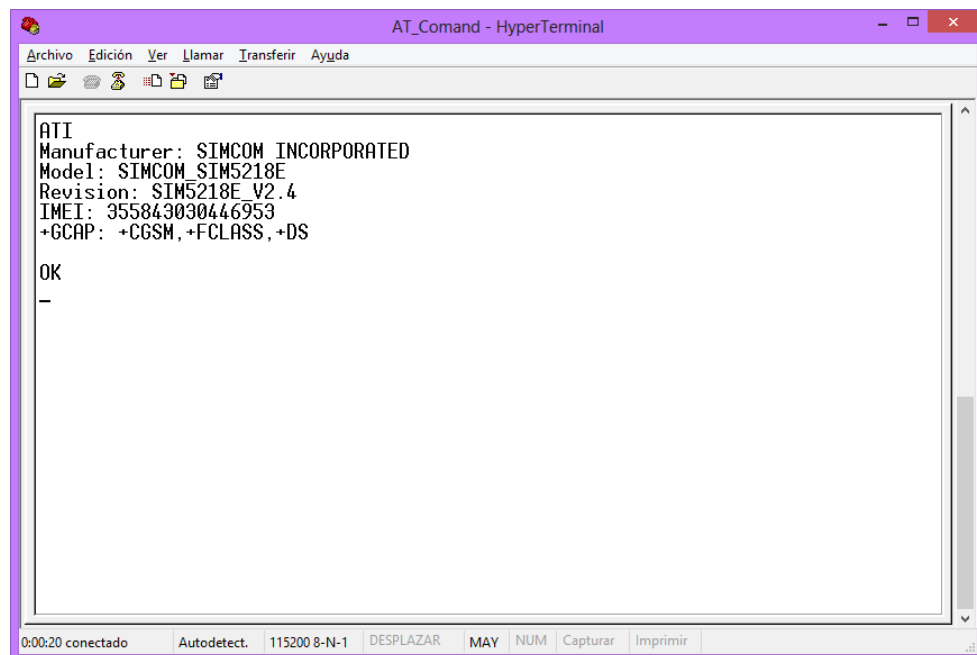


Figura 71: Información del Módulo 3G (HyperTerminal)

Control de módulo 3G mediante aplicación Android





Ejemplo de envío de SMS mediante comandos AT al número 65027... con el texto “prueba 1”. Primero se configura el modo texto con [AT+CMGF=1] se presiona Enter y luego se introduce el numero destino con [AT+CMGS=”650123456”] se presiona Ctrl+Z y aparece la llave donde se introduce el texto a enviar, y al terminar el texto, presionar Enter.

```
at+cmgf=1
OK
at+cmgs="65027[redacted]"
> prueba 1

+CMGS: 0

OK
```

Figura 72: Envío de SMS (Hyper Terminal)

I. Configuraciones

Configuración para comunicación entre el Módulo 3G y Android

Para establecer comunicación directa entre el Módulo 3G (sin Arduino) y la tableta Android es necesario utilizar un emulador de terminal que permita la comunicación con comandos AT, pero en la Play Store de Google no existe nada parecido, ni se han encontrado librerías que faciliten la comunicación, así que esta manera queda descartada.

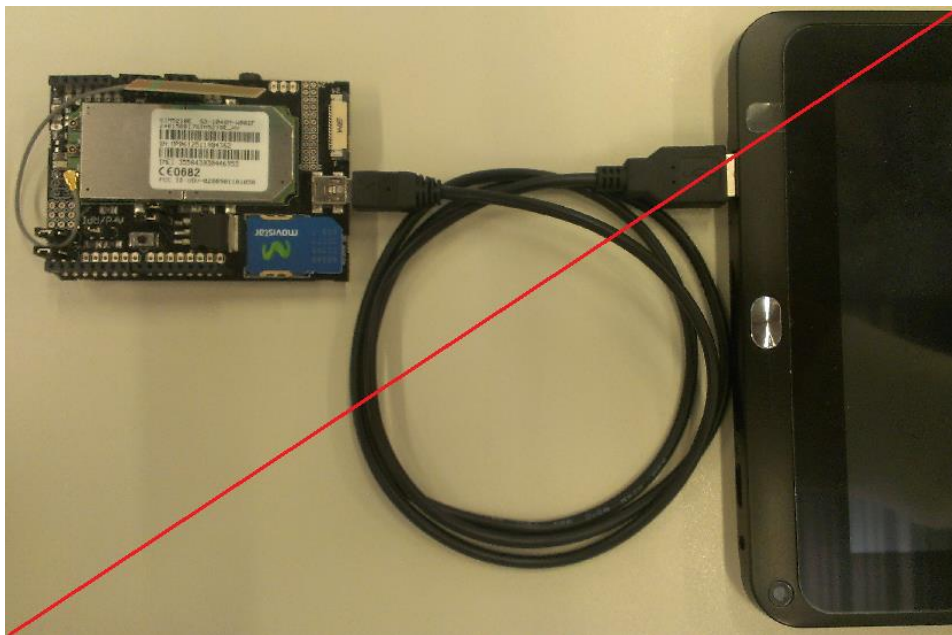


Figura 73: Conexión directa Módulo 3G-Tableta

Por tanto, la solución más sencilla es utilizar los conocimientos adquiridos en la parte de Arduino ya que se ha conseguido establecer una comunicación. Lo que se hará es insertar el Módulo 3G sobre Arduino (utilizándolo como Gateway) y trabajar con la comunicación serie con la cual sí que existen aplicaciones en el mercado.

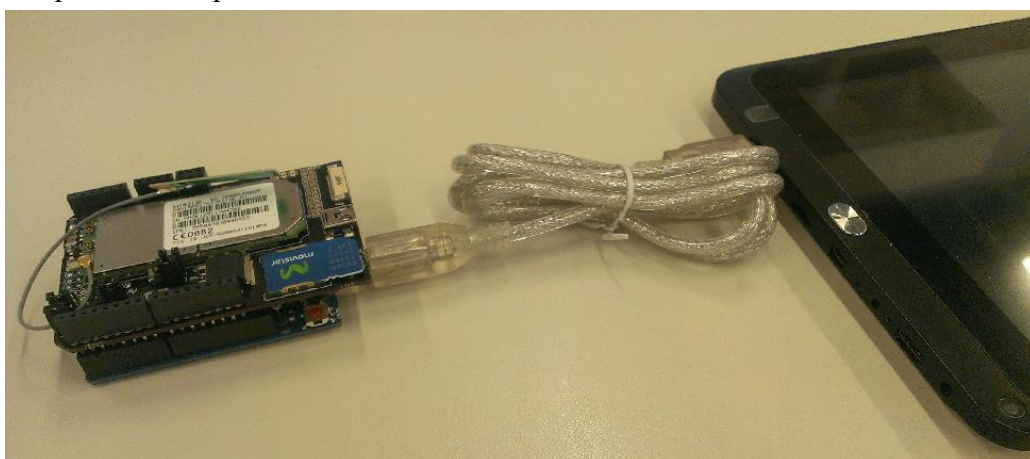


Figura 74: Conexión del Módulo 3G insertado en Arduino a la tableta

Para poder realizar la comunicación Módulo 3G - Tableta a través de la placa Arduino, utilizando la comunicación serie, se tienen que posicionar los jumpers (pequeñas piezas de plástico que efectúan un puente) en la posición correcta de modo que:

Control de módulo 3G mediante aplicación Android

- Los jumpers de comunicación serie recuadrados en rojo deben estar en la posición USB Gateway, de la otra forma es para correr con un Sketch de Arduino, sin estar conectado por cable a ningún dispositivo.
- El jumper de placa recuadrado en azul debe estar en la posición Ard, ya que estamos utilizando una placa Arduino, el Módulo 3G también es compatible con Raspberry Pi, de ahí la otra posición RPI.
- El jumper de energía recuadrado en naranja, en caso de utilizar una fuente de alimentación externa se debe colocar el jumper de energía en la posición correcta. Si el Módulo 3G se alimenta por Arduino, el jumper de energía debe estar en la posición “Arduino 5V”. Si el Módulo 3G se alimenta por el pin Vin input, el jumper de energía debe posicionarse en “Vext”.

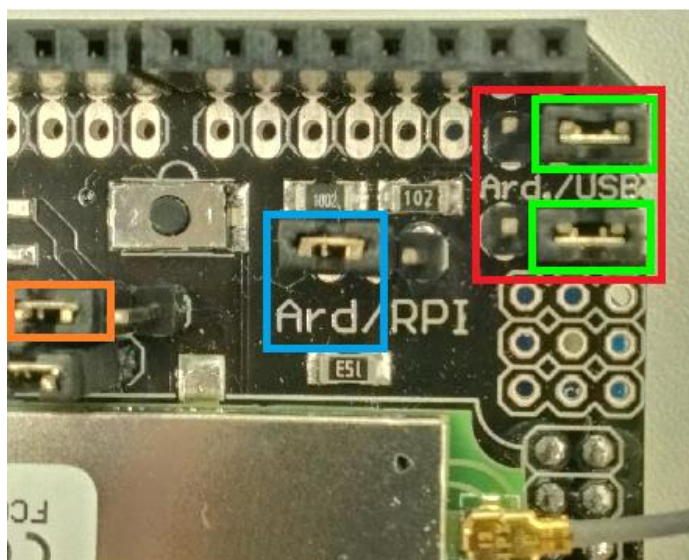


Figura 75: Detalle de los jumpers del Módulo 3G

En el siguiente diagrama se aprecian más claramente cómo tienen que disponerse las conexiones de los jumpers.

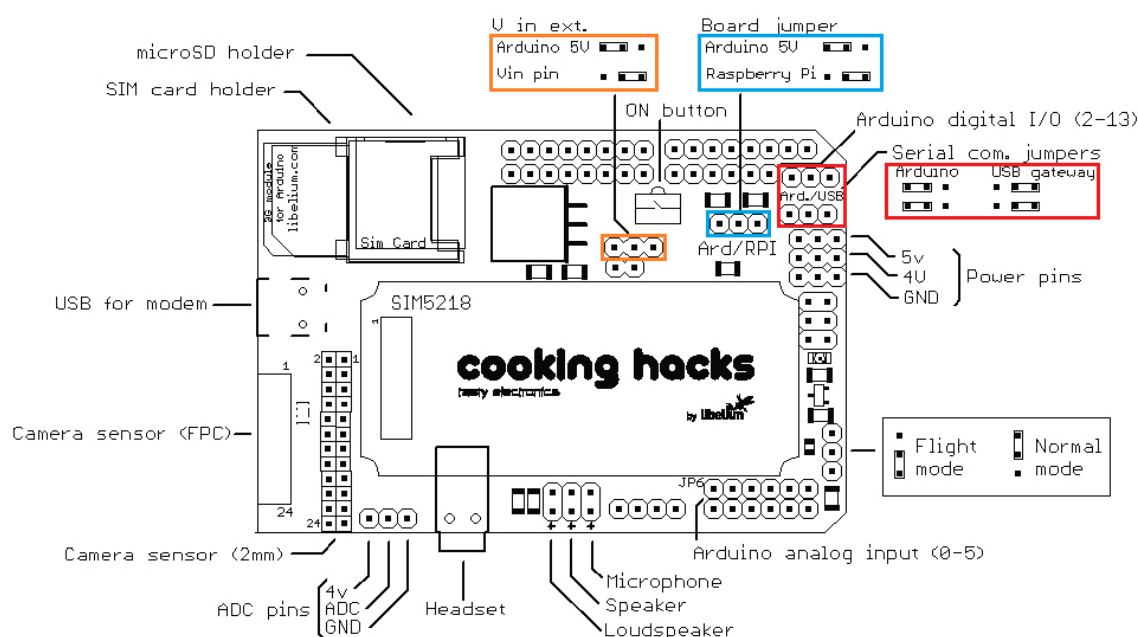


Figura 76: Diagrama del Módulo 3G



A continuación se debe escoger una de las siguientes opciones para utilizar Arduino de pasarela para la comunicación serie.

A. Retirando el chip microcontrolador ATmega de la placa Arduino.



Figura 77: Arduino UNO sin el microcontrolador

B. Realizar un puente en Arduino entre GND y RESET impidiendo que el microcontrolador se inicie.

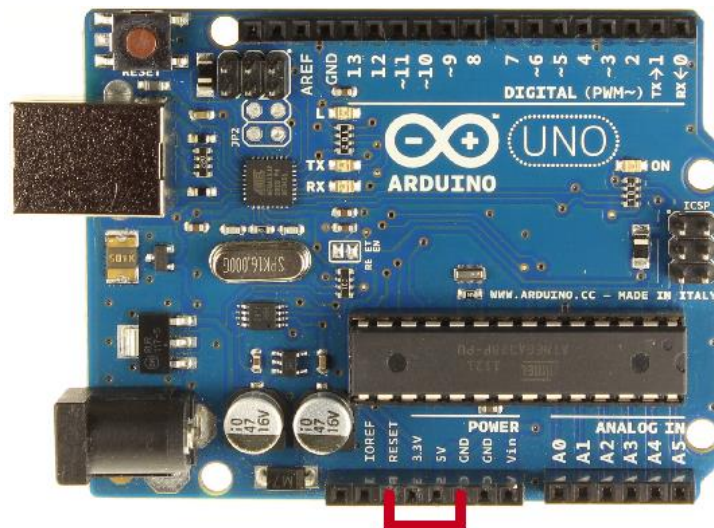


Figura 78: Detalle del puente en Arduino

C. Cargar un Sketch vacío en Arduino, esto impide que el gestor de arranque de la placa Arduino responda. Ésta ha sido la opción escogida por ser la más sencilla.

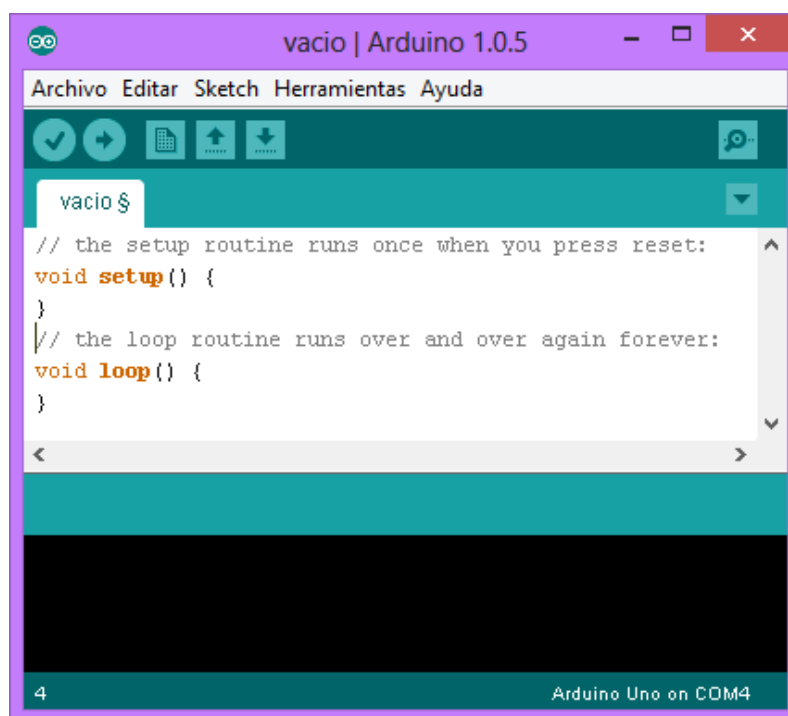


Figura 79: Sketch vacío

Una vez escogida una opción de las anteriores para que Arduino transmita la información al Módulo 3G, se debe colocar el Módulo 3G sobre Arduino.



Figura 80: Módulo 3G sobre Arduino

Y por último conectar Arduino a la tableta mediante el cable USB tipo A – tipo B.

Configuración del Módulo 3G para la conexión a red y la gestión de llamadas

Con el paquete del Módulo 3G se incluye entre otros elementos, un micrófono, un altavoz auricular los cuales se utilizarán para realizar las llamadas telefónicas, y una antena interna para cubrir las coberturas 2G y 3G.

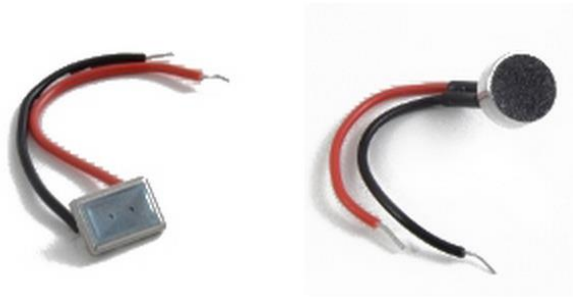


Figura 81: Micrófono y auricular



Figura 82: Antena 2G/3G

También es imprescindible disponer de una tarjeta SIM ya que ésta no se incluye con el paquete del Módulo 3G, puede ser de cualquier operador.

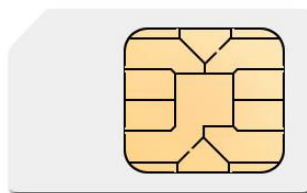


Figura 83: Tarjeta SIM



El micrófono y el auricular se conectan en los pines tal y como se muestra a continuación.

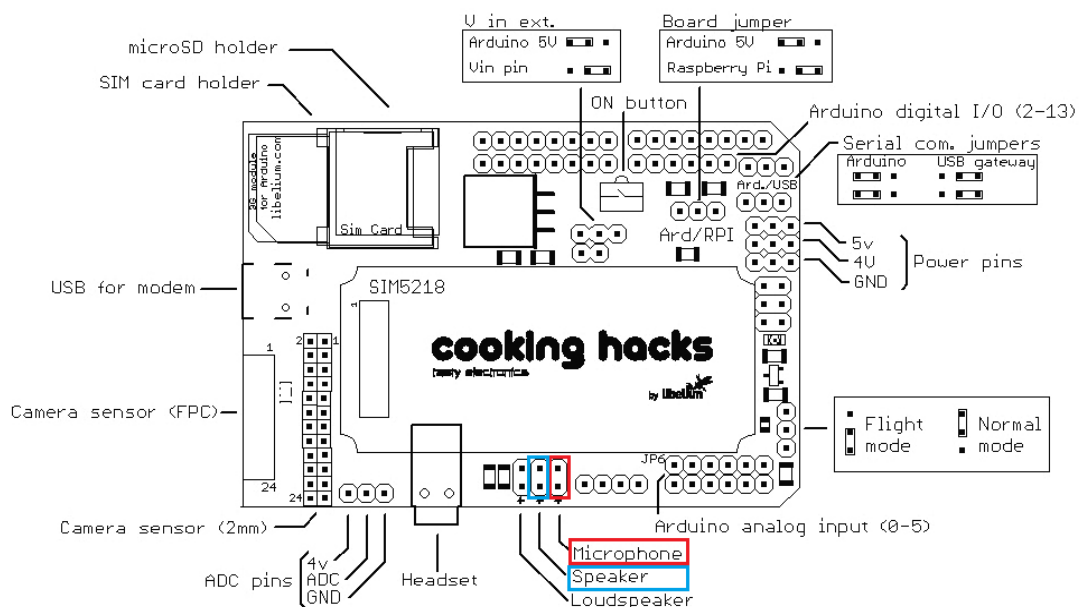


Figura 84: Diagrama Módulo 3G (Audio)

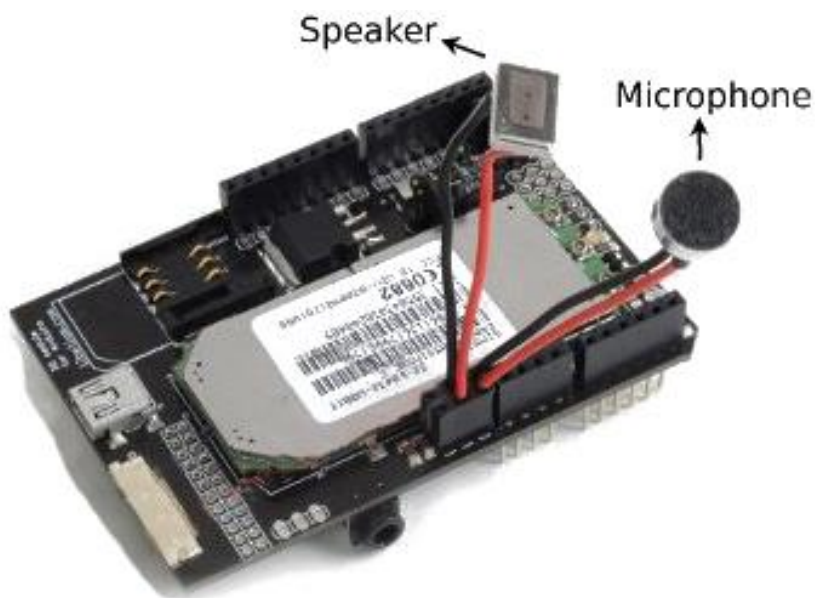


Figura 85: Módulo 3G con elementos de audio



La antena se conecta al módulo SIM5218 en el conector principal de la antena, ya que los otros conectores son para la antena del GPS y otra antena secundaria de red.

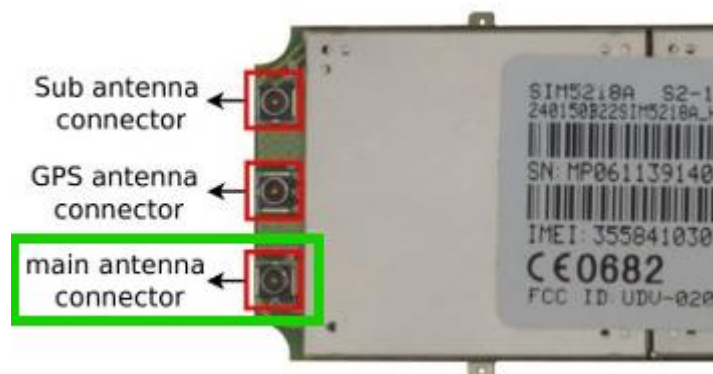


Figura 86: Conectores SIM3218 para antenas

La colocación de la antena queda como se muestra en la siguiente imagen.

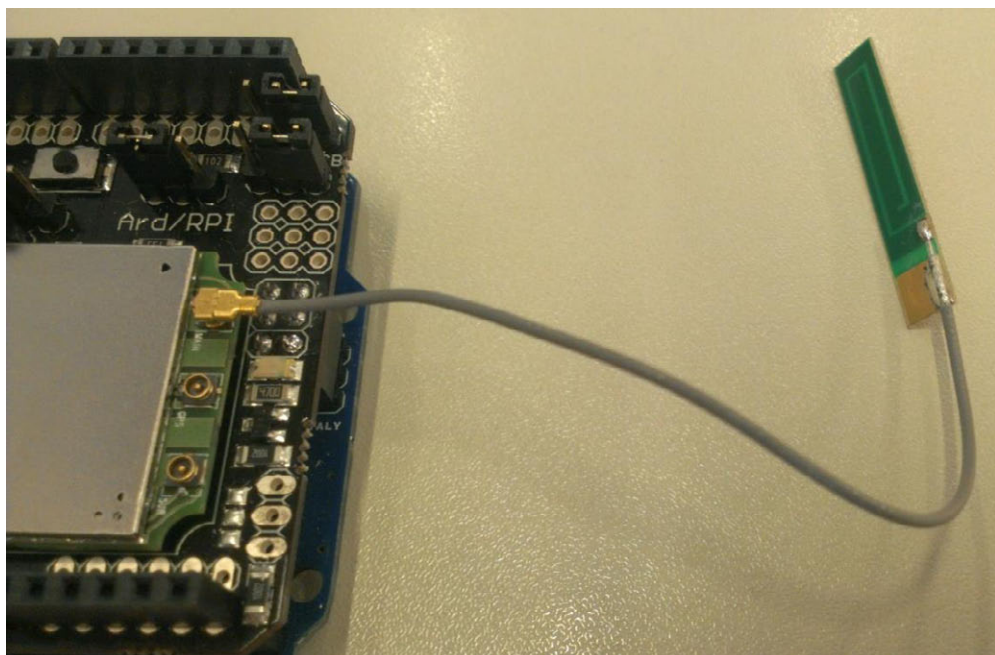


Figura 87: Antena enchufada al Módulo 3G



Para evitar tener los cables sueltos, que se puedan enganchar y tener una mayor comodidad a la hora de realizar pruebas con llamadas utilizando el micrófono y el auricular, se ha creado una estructura de cartón similar a un teléfono para fijar el Módulo 3G y la placa a él.

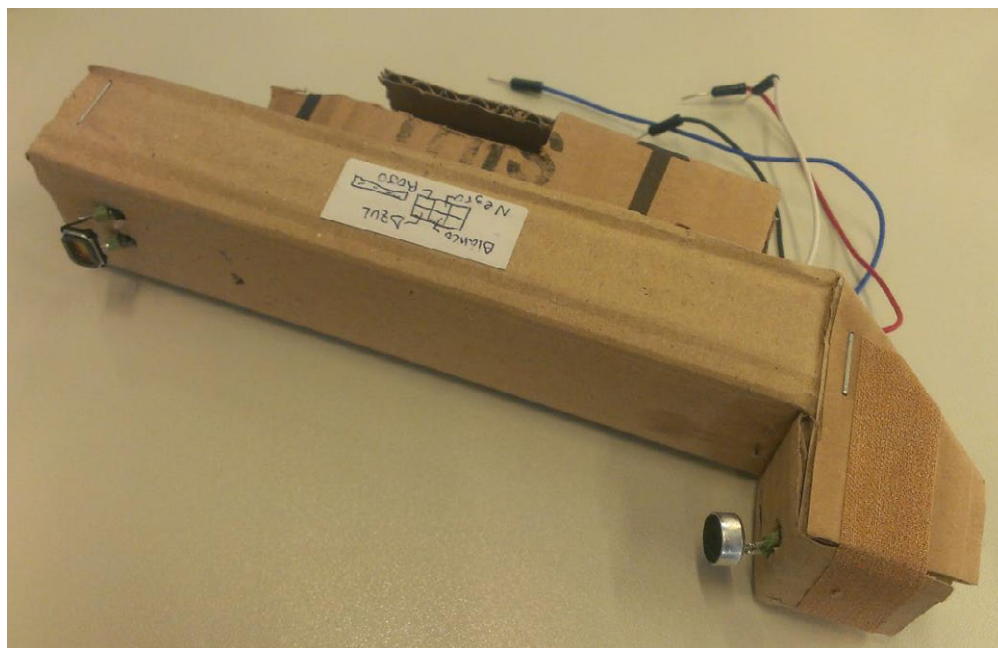


Figura 88: Teléfono de cartón

Las placas se introducen dentro del teléfono fabricado en cartón y queda todo más recogido y manejable para su funcionamiento.

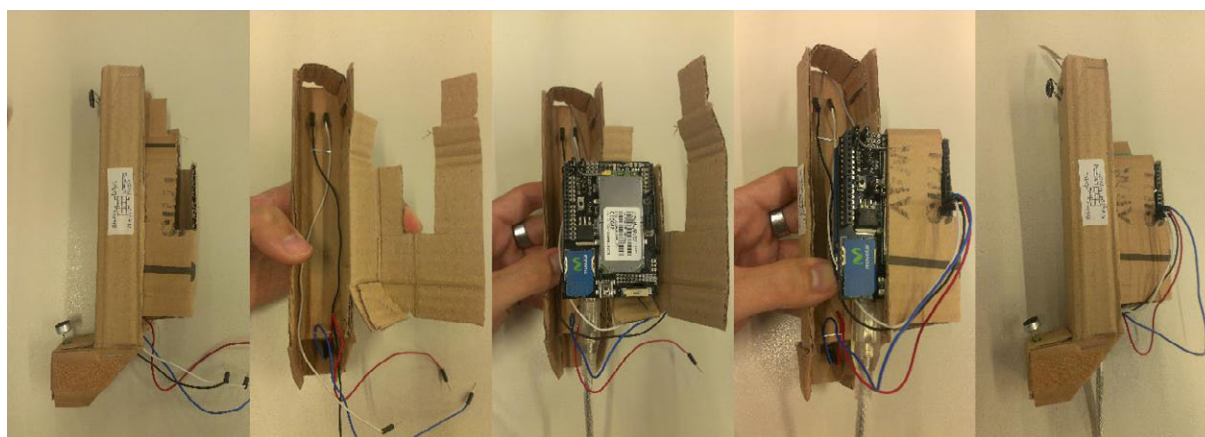


Figura 89: Secuencia de colocación del Módulo 3G en teléfono de cartón



J. Guía de comandos AT

En este apartado se exponen los comandos AT necesarios para manejar el Módulo 3G y ejecutar las funciones de realizar y recibir llamadas, enviar y recibir SMS, y descubrir las celdas vecinas de la red celular del sistema de telefonía móvil.

Esto es solo una muestra de lo que se puede llegar a hacer con los comandos AT, para conocer todos los comandos y todo el abanico de posibilidades existente consultar el manual de comandos AT para SIM5218 (23).

Consideraciones previas

Si se utiliza la librería Physicaloid se debe configurar la velocidad de transmisión a 9600.

Después de cada comando AT se debe enviar retorno de carro y salto de línea.

Cada respuesta recibida siempre termina con un “OK”.

Al final de cada descripción de comando se muestra el código que debería ir en la aplicación según la sintaxis de la librería Physicaloid, tal que así:

```
byte[] buf = "comando_AT".getBytes();  
mPhysicaloid.write(buf, buf.length);
```

Definiciones sintácticas

<CR> Carácter de retorno de carro. En Android “\r”.

<LF> Carácter de salto de línea. En Android “\n”.

<...> El texto encerrado entre llaves es un elemento sintáctico, las llaves en sí no aparecen en la línea de comandos.

Comandos AT generales

- **AT Verificación de la conexión**

AT es el comando más básico y es utilizado para comprobar que existe conexión, su respuesta es “OK” si no se recibe, es que no se ha establecido bien la conexión.

- **ATI Información de identificación de la placa**

ATI devuelve el fabricante, modelo, revisión, IMEI y lista de capacidades.

- **AT+IPREX Cambiar permanentemente la velocidad de transmisión (baud rate)**

AT+IPREX? Para saber qué velocidad está configurada actualmente.

AT+IPREX=<velocidad> Para fijar el valor de la velocidad de transmisión por ejemplo a una velocidad de 9600 bps: AT+IPREX=9600.





El Módulo 3G admite estas velocidades: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600, 3200000, 3686400, 4000000.

Se recalca que al emplear la librería Physicaloid es IMPRESCINDIBLE fijar la velocidad de transmisión del Módulo 3G a un valor bajo, a 9600 bps será suficiente. Si no se realiza esto, la comunicación serie funcionará mal.

- **AT+CNMP Selección de modo predeterminado**

El Módulo 3G soporta dos modos de comunicaciones móviles: GSM (2G) y WCDMA (3G).

AT+CNMP? para saber en qué modo se encuentra.

AT+CNMP=<modo> para cambiar el modo, donde los modos se introducen con su correspondiente número: 2 Automático, 13 sólo 2G y 14 sólo 3G. Por ejemplo AT+CNMP=13 (para sólo 2G).

- **AT+CPIN Introducir pin de tarjeta SIM.**

AT+CPIN=<pin> para introducir el pin de la tarjeta SIM, por ejemplo con un pin de 1234: AT+CPIN=1234.

Comandos AT para llamadas

- **ATD Iniciar llamada**

ATD para realizar una llamada telefónica de voz, se utiliza seguido del número de teléfono, terminado en punto y coma:

ATD650123456;

Sintaxis para Android:

```
byte[] buf = ("ATD"+numero+";" + "\r\n").getBytes(); //INICIAR LLAMADA.
```

- **ATA Descolgar**

ATA se usa a la hora de recibir llamadas y poder contestarlas. Se sabe que se está recibiendo una llamada cuando desde el Módulo 3G estamos recibiendo en Android el comando RING, envía un RING en cada tono automáticamente sin realizar ninguna solicitud.

```
byte[] buf = ("ATA" + "\r\n").getBytes(); //DESCOLGAR.
```

- **AT+CHUP Colgar**

AT+CHUP sirve tanto para colgar llamadas como para cancelar el marcado antes de que se descuelgue en destino.

```
byte[] buf = ("AT+CHUP" + "\r\n").getBytes(); //COLGAR.
```





Comandos AT para SMS

Para enviar un SMS antes se tiene que configurar el formato en modo "texto", y para leer, además, se tiene que seleccionar el lugar de almacenamiento de los mensajes.

- **AT+CMGF Selección de formato de SMS**

AT+CMGF=<modo> donde las opciones son 0=PDU o 1=texto, para leer y escribir un SMS se tiene que configurar el formato en modo texto de forma que AT+CMGF=1.

```
byte[] buf1 = ("AT+CMGF=1" + "\r\n").getBytes();
```

- **AT+CPMS Selección de almacenamiento de mensajes.**

Para leer SMS o eliminarlos, antes hay que escoger la memoria donde estén almacenados. Aquí los leeremos de la tarjeta SIM. Para seleccionar la tarjeta SIM como almacén de SMS se introduce AT+CPMS="SM","SM","SM" donde SM significa SIM. Devolverá una respuesta de este estilo: +CPMS:3,30,3,30,3,30 donde 3 es el número de SMS y 30 la capacidad de memoria de la SIM.

```
byte[] buf2 = ("AT+CPMS=\"SM\", \"SM\", \"SM\" + "\r\n").getBytes();
```

- **AT+CMGS Enviar SMS**

Una vez establecido el formato texto (AT+CMGF=1), se procede a introducir el número de teléfono destino de esta forma: AT+CMGS="650123456"<CR> y a continuación aparecerá una llave >, lo cual indica que se puede introducir el texto del mensaje y al final habrá que presionar Ctrl-Z.

```
AT+CMGS="650123456"<CR>
```

```
>Texto del sms<Ctrl-Z>
```

```
byte[] buf2 = ("AT+CMGS=\"\" + num + "\"\r" + "\r\n").getBytes();
```

```
byte[] buf3 = ( sms + "\032" + "\r\n").getBytes(); // 032 = CTRL-Z
```

- **AT+CMGR Leer SMS de uno en uno**

Una vez establecido el formato texto (AT+CMGF=1) y escogido la memoria (AT+CPMS="SM","SM","SM") se escoge el SMS a leer de la lista seleccionando la posición empezando por 0. Si se ha obtenido +CPMS:3,30,3,30,3,30 y se quiere leer el último SMS recibido se tiene que introducir AT+CMGR=2. Y si se quiere leer el primero AT+CMGR=0.

```
byte[] buf3 = ("AT+CMGR=0" + "\r\n").getBytes();
```

- **AT+CMGL Listar SMS**

Otra forma de leer los SMS es listarlos todos en pantalla: AT+CMGL="ALL"

```
byte[] buf3 = ("AT+CMGL=\"ALL\" + "\r\n").getBytes();
```





Comandos AT para información de celdas

Para mostrar un ejemplo de la información que se puede llegar a extraer con el Módulo 3G de la red de comunicaciones móviles con los comandos AT, se va a utilizar un comando que devuelva información de la celda que actualmente brinda el servicio y de las celdas vecinas. AT+CCINFO.

- **AT+CCINFO Show cell system information**

Antes de ejecutar el comando se tiene que configurar la selección de red a GSM ya que este comando solo trabaja con este modo, para comparar celdas en WCDMA (3G) no existen comandos.

El comando solicita información del sistema de celdas tanto vecinas como de la servidora, en GSM.

Se obtiene de respuesta para cada celda los campos:

- CCINFO: tipo de celda, servidora (SCELL) o vecina (NCell1) (NCell2)...
- ARFCN (Absolute Radio-Frequency Channel Number): Canal radio asignado.
- MCC (Mobile Country Code): Código del país. España 214.
- MNC (Mobile Network Code): Código del operador. Movistar 07, Vodafone 01, Orange 03.
- LAC (Location área code): Código de área de localización.
- ID (Cell Identifier): Identificador de celda.
- BSIC (Base Station Identification Code): Número identificativo de cada estación base.
- RXlev: Intensidad de señal recibida en dBm.
- TA (Timing Advance): Parámetro de tiempo que ayuda a la sincronización entre el terminal móvil y la estación base.
- C1: Coeficiente para la selección de estación base.
- C2: Coeficiente para la re selección de celda.
- TXPWR: Potencia de transmisión del equipo de usuario en dBm, si no hay transmisión, el valor es 0.

En el código Android aparecerá de esta forma:

```
byte[] buf = ("AT+CCINFO" + "\r\n").getBytes();  
mPhysicaloid.write(buf, buf.length);
```

- **AT+CPSI Solicitud de información del equipo se usuario.**

Éste es otro comando con el que obtener información de la celda a la que se está conectado actualmente. Si se está conectado a 2G devolverá una información, y si es a 3G devolverá otra.

```
byte[] buf = ("AT+CPSI?" + "\r\n").getBytes();
```

En 2G:

- System Mode: “NO SERVICE”, “GSM” o “WCDMA”.
- Operation Mode: “Online”, “Offline”...
- MCC: Código del país.





- MNC: Código del operador.
- LAC: Código de área de localización.
- Cell ID: Identificador de celda.
- Absolute RF Ch Num: Canal radio asignado.
- RxLev: Intensidad de señal recibida en dBm.
- Track LO Adjust
- C1: Coeficiente para la selección de estación base.
- C2: Coeficiente para la re selección de celda.

En 3G:

- System Mode: “NO SERVICE”, “GSM” o “WCDMA”.
- Operation Mode: “Online”, “Offline”...
- MCC: Código del país.
- MNC: Código del operador.
- LAC: Código de área de localización.
- Cell ID: Identificador de celda.
- Frequency Band: Banda de frecuencias.
- PSC (Primary Synchronization Code): Código de sincronización primario.
- Freq: Frecuencia del enlace descendiente
- SSC (Secondary Synchronization Code): Código de sincronización secundario.
- EC/IO: Valor de EC/IO medida de la evaluación y las decisiones de CDMA y UMTS.
- RSCP (Received Signal Code Power): Potencia de la señal recibida.
- Qual: Valor de la calidad para la selección de estación base.
- RxLev: Intensidad de señal recibida en dBm.
- TXPWR: Potencia de transmisión del equipo de usuario en dBm, si no hay transmisión, el valor es 500.







Referencias

Todos los enlaces han sido consultados en Abril de 2014.

1. Internet de las cosas. [En línea] http://en.wikipedia.org/wiki/Internet_of_Things.
2. Evans, Dave. Internet of things how the next evolution of the internet is changing everything. Cisco. [En línea] https://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf.
3. Web oficial de Union Internacional de Telecomunicaciones. [En línea] <https://www.itu.int/>.
4. Web oficial 3rd Generation Partnership Project. [En línea] <http://www.3gpp.org/>.
5. Mapa cobertura LTE. XatakaMovil. [En línea] <http://www.xatakamovil.com/movistar/mapa-cobertura-lte-y-comparativa-del- calendario-de-despliegue-en-espana>.
6. Estándares de redes celulares. Wikipedia. [En línea] http://en.wikipedia.org/wiki/Template:Cellular_network_standards.
7. Web oficial de Android para desarrolladores. [En línea] <http://developer.android.com/>.
8. Entorno de desarrollo Eclipse para Android. Web oficial de Android para desarrolladores. [En línea] <http://developer.android.com/sdk/index.html>.
9. Proyecto Physicaloid. GitHub. [En línea] <https://github.com/ksksue/PhysicaloidLibrary>.
10. Web oficial librería Physicaloid. [En línea] <http://www.physicaloid.com/>.
11. Proyecto USB Serial for Android. GitHub. [En línea] <https://github.com/mik3y/usb-serial-for-android>.
12. Proyecto USB Serial for Android. Google group. [En línea] <https://groups.google.com/group/usb-serial-for-android>.
13. API para USB Host. [En línea] <http://developer.android.com/intl/es/guide/topics/connectivity/usb/host.html>.
14. Web oficial Arduino. [En línea] <http://www.arduino.cc/>.
15. Lenguaje Arduino. Arduino. [En línea] <http://arduino.cc/es/Reference/HomePage>.





16. Manual del Módulo 3G. Cooking Hacks. [En línea] <http://www.cooking-hacks.com/documentation/tutorials/arduino-3g-gprs-gsm-gps>.
17. Web oficial Cooking Hacks. [En línea] <http://www.cooking-hacks.com/>.
18. Comunicación serie. [En línea] <http://galaxi0.wordpress.com/el-puerto-serial/>.
19. Puertos Serie. [En línea] <http://c2.com/cgi/wiki?SerialPorts>.
20. Web oficial Point of View. [En línea] <http://www.pointofview-online.com/>.
21. Google Play. [En línea] <https://play.google.com/>.
22. Manual de comandos AT V1.21. SIM Com. [En línea] http://www.cooking-hacks.com/skin/frontend/default/cooking/pdf/SIM5218_AT_command_manual.pdf.
23. Manual de comandos AT V1.40. SIM Com. [En línea] http://microchip.ua/simcom/WCDMA/SIM5218/SIMCOM_SIM5218_Serial_ATC_EN_V1.40.pdf.
24. USB OTG. Wikipedia. [En línea] http://es.wikipedia.org/wiki/USB_On-The-Go.
25. Android USB Host + Arduino. [En línea] <http://android.serverbox.ch/?p=549>.
26. Modo USB Host y Accesory. Android Developers. [En línea] <http://developer.android.com/guide/topics/connectivity/usb/index.html>.
27. SDK Android. Android Developers. [En línea] <https://developer.android.com/intl/es/sdk/index.html>.
28. Tutorial Android. Android Developers. [En línea] <http://developer.android.com/training/index.html>.
29. Instalación SIM5218. Cooking Hacks. [En línea] http://www.cooking-hacks.com/skin/frontend/default/cooking/pdf/3G_modem_tutorial.pdf.
30. Drivers SIM5218. Cooking Hacks. [En línea] http://www.cooking-hacks.com/skin/frontend/default/cooking/pdf/WindowsDrivers_SIM5218_2.4.00.rar.
31. Descarga HyperTerminal. [En línea] <https://app.box.com/shared/8vehulvb7u>.
32. LTE no es 4G. Xataka. [En línea] <http://www.xataka.com/otros/lte-y-lte-advanced-cual-de-ellos-es-realmente-4g>.

